SUDAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

COLLEGE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

SCRUM DEVELOPMENT PLATFORM

OCTOBER 2015

THESIS SUMITTED AS A PARTIAL REQUIREMENTS 0F B.Sc. (HONOR) DEGREE IN SOFTWARE ENGINEERING

SUDAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

COLLEGE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

SOFTWARE ENGINEERING DEPARTMENT

SCRUM DEVELOPMENT PLATFORM

Prepared By:

Ahmed Taj elsir Ali

Ayman Adam Dawood

Hammad Dafer Mohammed Osman

Supervisor:

Hana Al-tayb

THESIS SUMITTED AS A PARTIAL REQUIREMENTS 0F B.Sc. (HONOR) DEGREE IN SOFTWARE ENGINEERING

Supervisor Signature: **Date:**

……. **15/10/2015**

October 2015

الآيـــة

قال تعالى:

﴿ وَمَا تَوْفِيـقِي إِلَّا بِاللَّهِ
عَلَيْهِ تَوَكَّلْـتُ وَإِلَيْهِ أُنِيبُ ﴾

صدق الله العظيم

سورة هود الآية ﴿88﴾

الحمـد لله

اللَّهُمَّ لك الحَمْدُ كَمَا حَمِدْتَ نَفْسَك

في أُمِّ الكِتَابِ والتَّوْرَاةِ والإِنْجيْلِ والـزَّبُورِ والفـرْقَان

اللَّهُمَّ لك الحَمْـــــــدُ أَكْــمَلُهُ،

ولك الثَّـــــنَاءُ أجْـــــمَلُهُ،

ولك القـــــــــوْلُ أبْلَغُهُ،

ولك العِلْـــــمُ أحْـكَمُهُ،

ولك السُّلْـــطَانُ أقْوَمُهُ،

ولك الجَلالُ أعْظَمُهُ

DEDICATION

We would love to thank everyone who encouraged us to achieve this tremendous task

Especially our parents who believed in us, and that we could actually achieve such a thing

All our friends and colleagues for always being there for us in time of need it has truly been an honor to have colleagues of such caliber

Our families, thank you for believing in us, please do not ever doubt our dedication and love for you.

We would love to dedicate a second special thanks to our parents as we could never express our gratitude towards them no words can give them justice

ACKNOWLEDGEMENT

We thank Almighty Allah, for giving us the courage and determination, as well as guidance to conduct this research.

We would like to acknowledge our supervisor Ms. Hanaa for all her effort and guidance and her assistance.

ABSTRACT:

In the past couple of years Agile development has gotten very famous because of its characteristics and ability to frequently deliver satisfactory software. Making it take its place on top of other methods, making companies want a slice of that cake so there have been attempts to distribute agile development, although agile methods prefer direct interaction rather than tools, it has been proved to have the capability to be distributed.

Scrum is one of the most commonly used Agile Development Methods used in the industry, it revolves around frequent delivery and the ability to effectively respond to change this ability is called “requirements churn”. It has a set of defined roles each with its own responsibility and corresponding tasks, the Product Owner, Scrum Master and Development Team. It’s well known for its simplicity and easy implementation making it a wide spread method.

International Companies wish to use their resources which are located across the globe in a successful agile manner to utilize its skilled manpower in the most efficient way.The project aims to create a platform to enable and enhance development for agile projects, it attempts to provide everything that an agile project will require throughout its development.

The platform targets both remote and local teams as it enables plenty of communication features and high project visibility, the high project visibility is established through various platform components whose coherent work will allow that level of visibility. It provides methods for communication in manners like video conferencing and chatting, also the task assignment and task creation all coupled with a coding environment and version control, when all of these work together they provide full project visibility from one place as to what tasks are left, what are the bugs, what are the missing features and the current source code.

المستخلص:

في الأعوام الماضية اشتهر التطوير المرن جدا بسبب مميزاته وقدراته على اصدار برمجيات مرضية للعملاء، مما ساعد في تصدره على رأس الطرق الاخرى. لذا سعت شركات البرمجيات الدولية الكبرى الي استخدام مواردها التي تقع في جميع أنحاء العالم بطريقة التطوير المرن للاستفادة من المبرمجين المهرة حول العالم بطريقة أكثر فعالية .

الاسكرم من اكثر طرق التطوير المرن استخداما في المجال ، فهو يتمحور حول التسليم المتكرر والقدرة على الاستجابة للتغيرات . لديه العديد من الادوار كل بمسؤوليته ومهامه ، (صاحب المصلحة ، منظم الفريق ، فريق التطوير) . ويعرف بالبساطة وسهولة التطبيق مما جعله الاكثر انتشارا .

برغم اثبات فعالية التطوير المرن في البيئات المتباعدة جغرافيا الا انه يفضل التفاعل المباشر بدلا من الأدوات، لذا يهدف هذا المشروع إلى إنشاء منصة لتمكين وتعزيز تطوير المشاريع التي تستخدم التطوير المرن وخصوصا الاسكرم ، محاولا تقديم كل ما تحتاجه المشاريع على مدى فترة تطويرها.

المنصة تحاول مساعدة فرق التطوير ان كانوا يعملون بالتطوير عن بعد او كان الفريق متمركز في مكان واحد، تسهل المنصة عملية تواصل المعلومات من خلال مميزات النظام مما يقدم وضوح عال لوضع المشروع .

هذا الوضوح يتم بسبب اجزاء النظام التي تتكامل مع بعضها لتقديم الصورة الكاملة لمديري المشاريع، و اصحابها والمطورين انفسهم.

يقدم النظام وسائل للتواصل بطرق عده مثل اجتماعات الفيديو و الدردشة بين اعضاء المشروع، ايضا انشاء وادارة المهمات مع بيئة التطوير بالاضافة الي التحكم باصدارات المشروع.

عندما تلتقي كل هذه المميزات معا تقدم وضوح عال للمشروع و كل هذا من خلال المنصة نفسها، و ذلك يضمن معرفة ما المهمات المتبقية، و ما المشاكل التي تواجه تقدم المشروع، كل ذلك بالاضافة الى الشفرة المصدرية للمشروع.

TABLE OF TERMS:

#	Term	Description
1	ASD	Agile Software Development
2	TTM	Time To Market
3	ADM	Agile Development Methods
4	DSDM	Dynamic System Development Method/Driving Strategy Developing More
5	FDD	Feature Driven Development
6	DSD	Distributed Software Development
7	DAD	Distributed Agile Development
8	GSD	Global Software Development
9	XP	Extreme Programming
10	HTML	Hyper Text Markup Language
11	CSS	Cascading Style Sheets
12	SIL OFL	Sil Open Font License
13	MIT	Massachusetts Institute of Technology
14	Ajax	Asynchronous Java Script and XML
15	IDE	Integrated Development environment
16	HTTP	Hyper Text Transfer Protocol
17	MVC	Model View Controller
18	WSGI	Web Server Gateway Interface
19	API	Application Programming Interface
20	SHA	Secure Hash Algorithm
21	AES	Advanced Encryption Standard
22	DVCS	distributed version control system
23	VCS	Version Control System
24	SQL	Structured Query Language
25	ORDBMS	Object-relational Database Management System
26	SIAB	Shell In A Box
27	WebRTC	Web Real Time Communication

TABLE OF FIGURES:

TABLE OF FIGURES:

TABLE OF TABLES:

TABLE OF CONTENTS:

CHAPTER ONE
INTRODUCTION

1.1 INTRODUCTION:

There have been many early attempts to improve software quality focused on better methods of defining, detailing and dealing with Software Requirements. Most of these methods were iterative and incremental, these methods were able to change, bend and adapt to requirement changes which gave them an agile form and resilience to modification. These methods were called agile methods.

These methods trace back to 1975 and were introduced as Adaptive Software Development Process, and has evolved through time to achieve the flexibility that Agile Methodologies have achieved these methodologies focused on a different aspect of development than the usual water-fall heavy methods; such as Individuals and Interactions over process and tools, a working software rather than a Comprehensive documentation, heavy customer collaboration over contract negotiation and most importantly the ability to respond to change rather than following a strict plan [1].

Agile Software Development (ASD) has become a very popular stand for a lot of the software industry. In a lot of cases it causes more productivity and increases quality which results in higher stakeholder satisfaction also it heavily decreases Time to Market (TTM) [1] [2] [3].

ASD achieves all of the above by changing the principles of the conventional software requirement making it a lot more tolerant to change. The principles in Agile are more close & personal than formal & strict such as daily cooperation between clients and developers and focusing on face to face communication. All the other principles have the same idea in mind when it comes to team management and software delivery

ASD has gained much attention in the past couple of years, specially in the Software Engineering community, this is due to its resilience and ability to main software quality, Some of the most famous Agile Development Methods (ADM) are: Extreme Programming (XP), Scrum, Dynamic System Development Method (DSDM), all of the Crystal Family methods and Feature Driven Development (FDD). These methodologies all differ in their implementation of the ASD concepts but at the heart of them all the core values still remain.

Scrum is one of the most popular ADM as it holds true to the core values of ASD, it also adds it's own flavor to them by introducing things such as Product Backlog and Sprint Backlog and one of it's most important features is it's burn down chart which aids in calculating TTM, also it separates roles into 3 mains roles which are the Product Owner, The Scrum Master and The Developer. Also it has the ability to support distribution because of it's characteristics.

Distributed Software Development (DSD) enables teams to be in various locations operating remotely from each other, making up a network of sub teams, the teams could be from the same organization or a part of an outsource collaboration. This leads to problems in communication such as the in-ability to meet face to face relying on technology to facilitate coordination and communication.

Distributed Agile Development (DAD) has become a focus point for both industry and academia as Global Software Development (GSD) is becoming more mainstream than ever. However, agile development methodologies have always been designed for collocated teams making them harder to apply in DAD.

In the recent years, geographic distribution of development teams has been very common, enterprises tend to take advantage of skilled developer all of the world, and the main objective is to optimize resources to develop high quality software. Software Factories are therefore organizational structures which automate parts of software development by imitating those industrial processes that were originally linked to more traditional sectors such as those of the automobile and aviation industries, promoting re-usability of knowledge, architecture and components.

Scrum is commonly co-located teams working together to achieve their assigned tasks, so why is it capable of distribution?

Scrum has very adequate management in its practice and it's has a high level of co-ordination throughout team members, its ability to manage complexity excels it also it's scalability factor from single processes to entire projects make it a great candidate.

1.2 PROBLEM STATEMENT:

Enterprises have a lot of resources available and multiple teams available for the development of the software, the utilization of these resources is a big benefit for the company, having developers work round the clock to deliver software faster, but the Scrum methodology requires heavy synchronization and high project visibility for it to be accomplished and to achieve its task, problems arise when a bigger team needs to achieve those requirements and those problems become more significant when Scrum is attempted to be applied to distributed teams. The realization of the above requirements become a lot harder and difficulty faces the process completion.

Some of the many hardships form in communication between teams where they aren't following the same timezone and have problems co-ordinating, such as its night in one country and day in the other. Another problem is being able to view the code and having high visibility of the project's progress.

1.3 AIM:

Applying Agile practices to achieve higher customer satisfaction while utilizing enterprise’s global resource in the process via a platform that enables dispersed teams to function at a higher. The platform will aim to improve communication between teams and allow them to retain the core values of agile development even with distributed teams, also it will enable them to share the code for rapid development, while providing tools to easily develop applications, task management is also one of the many important features that scrum development requires and the platform provides as one of its services.

1.4 OBJECTIVES:

The major objectives for this research are:

- Management system.
- Version control system.
- Integrated development environment.
- Communication system.

- Effective control.
- Increase overall productivity.
- Cost savings.
- Access to large multi skilled workforces.
- Reduced time to market.
- Proximity to market and customer.
- Innovation and shared best practice.
- Resource allocation.
- Improved task modularization.
- Reduced coordination cost.
- Increased team autonomy.
- Formal record of communication.
- Improved documentation.
- Continuous integration
- Clearly defined processes.
- Code sharing.

1.5 SCOPE:

A management system for organizing and managing projects from things as simple as task assignment reaching up to sprint scope and entire project visibility, also including an interface with a popular version control system to make sure no piece of code is lost and regression in the project is easily done in addition to shared code and an IDE to develop with a specialized server running the code to take stress of developers machines, not to forget means of communication between team members, all these facilities are to offer any assistance a single platform could help a development team.

1.6 KEY RESEARCH QUESTIONS:

How can manage all tasks in Distributed team?

How can integrated development environment?

How can manage Communication in Distributed team?

1.7 STRUCTURE:

This is research is divided into six chapters include:

- Chapter 2: describes agile methods (specifically scrum) and distributed agile development.
- Chapter 3: describes tools that used during analysis and implementation
- Chapter 4: describes the analysis and design.
- Chapter 5: describes the implementation and testing of the platform.
- Chapter 6: Conclusions and Recommendations.

CHAPTER TWO
LITERATURE REVIEW

2.1 INTRODUCTION:

Interest in Agile Development Methodologies (ADM) has increased over the past few years, for numerous reasons such as higher customer satisfaction which has made it a leading methodology recently. Agile methodologies have been developed to overcome the limits and problems with the traditional software developments approaches, this for it to be adopted as standard in many leading companies.

Agile software development has gained acceptance in the mainstream software development community. Many surveys has shown that agile teams often are more successful than traditional ones [1] [4] [3].

Several other studies demonstrated 60% increase in productivity, quality and improved stakeholder satisfaction [3] [2], 40% faster time-to-market and 60% and 40% reduction in pre/post-release defect rates compared to the industry average [2].

2.2 AGILE METHODOLOGIES:

There are a large number of software development methods that have been introduced to become a part of the ADM umbrella. These methodologies include, Extreme Programming (XP) [5], Scrum software development [6], Driving Strategy Delivering More (DSDM) [7], Fetch Driven Development (FDD) [8], Crystal Family [8] and many other well-known methods.

A survey [2]has been conducted, its information was collected from 91 countries it shows that the most popular methods are Scrum (58%), XP (4%), and Scrum/XP Hybrid (17%). The 37% of respondents worked in distributed agile environments [2].

Here we will discuss the most commonly used agile methodology (scrum) because there are many papers and studies show that the team can be distributed; also we show that it's more popular than the other agile methods.

There are numerous agile development methods introduced and actually used in software in the industry, these methods include [8]:

- Adaptive Software Development.
- Agile Modeling.
- Crystal Family of Methodologies.
- Dynamic System Development Method.
- Extreme Programming [2] [5].
- Feature driven Development.
- Pragmatic Programming.
- Scrum [2] [6].

2.2.1 TYPE OF AGILE METHOD:

In this section, the researchers attempts to discuss agile methodology available and used in software industry. The most popular agile methods is: Scrum [6] [9] [10] [11] [12] [6] [13] [14].

2.2.1.1 SCRUM:

The most popular of agile methodologies Scrum, it is a frame work for organizing and managing work. The Scrum framework is based on a set of values, principles, and practices that provide the foundation to which your organization will add its unique implementation of relevant engineering practices and your specific approaches for realizing the Scrum practices [10].Scrum Roles: Scrum development efforts consist of one or more Scrum teams, each made up of three Scrum roles: product owner, Scrum Master, and the development team [6] [9] [10] [12] [9] [13].

2.3 SCRUM PRACTICES:

Scrum defines three main practices to be followed in the development process. Figure 1.illustrates Scrum Practices.

Product Backlog:

It's an order list came from product owner with integration of input from scrum team and the other stakeholder, it contains features required to satisfy the product owner requirements and it use to manage and control the team work. And it also can contains new features, changes to existing features, defects needing repair, technical improvements, and so on [9] [10] [11] [12] [13].

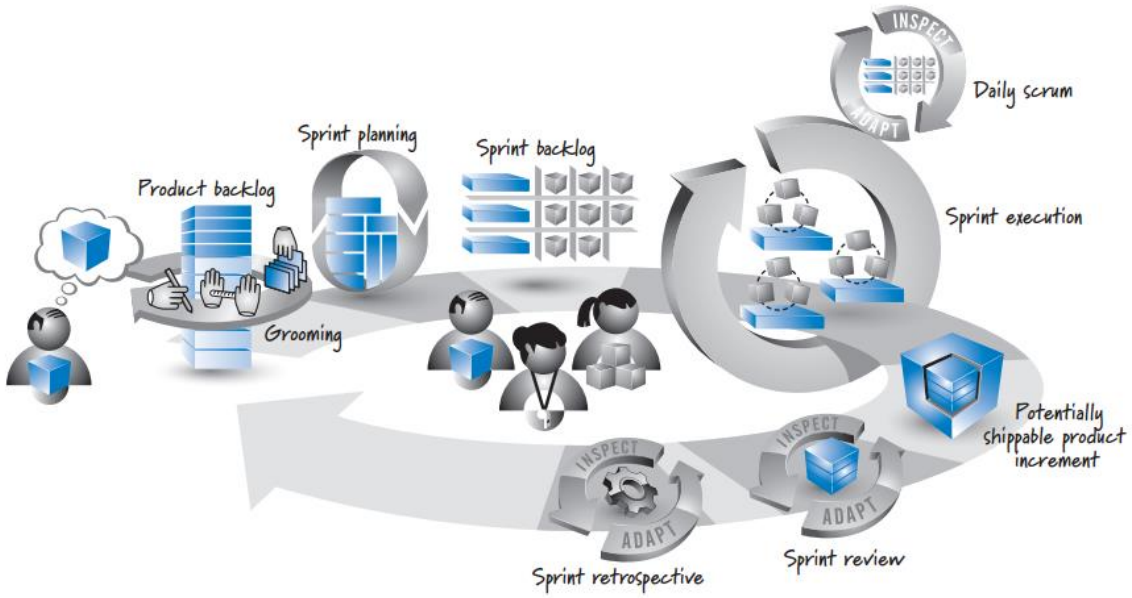


Figure 1 Scrum Practices *[10]*

Sprints:

Scrum team are working in an iterative manner for up to a month per iteration the iteration called Sprint , every sprint is done a function or component of product should be completed and my deliver [15]. Every sprint has a fixed start and end date and should have the same duration [6] [9] [10] [12] [13] [14].

Sprint Planning:

There is meeting held with the development team, management, and the Product Owner. To determine which the most important feature or items to produce in the next sprint is called sprint planning Meeting [10] [12] [15] [6] [14] [13].

Sprint Execution:

Once the Scrum team finishes sprint planning and select the next features to implements the development team Under the leadership of scrum master Prepare all the necessary requirements to get the selected feature done, where “done” means there is a high degree of confidence that all of the work necessary for producing good-quality features has been completed [10] [12] [6] [13] [14].

Daily Scrum:

This meeting is also called stand up meeting , and it hold every day for 15 minutes , and every team member should answer the underline three questions:

- What have you done since the last meeting?
- What will you do between now and the next meeting?
- What got in the way of doing your work?

To achieve the progress of the project and the points that have been accessed [10] [11] [9] [15].

Sprint Review:

when completing sprint, a sprint review takes place to review progress, display features to the customer, management, users and the Product Owner and review the project from a technical perspective [10] [6] [14] [15].

Sprint Retrospective:

After sprint review and before the next sprint planning the focus is on the continuous process improvement necessary to help a good Scrum team become great. At the end of a sprint retrospective the Scrum team should have identified and committed to a practical number of process improvement actions that will be undertaken by the Scrum team in the next sprint [10] [15].

let's go to the diagram and summarize it by starting on the left side Through a clockwise direction ,on the first the product owner has a clear vision of what he want to create, then group the stakeholder needs and break down into a set of features and collect , prioritized them in a list called product backlog.

the first sprint start by selecting the most important feature by the product owner then holding the sprint planning meeting, after that the sprint execution is start (the development during the work), and the sprint ends with sprint review and sprint retrospective [10] [15] [12] [6] [13].

To acquire confidence that the development team has made a reasonable commitment, the team members create a second backlog during sprint planning, called the sprint backlog. The sprint backlog describes, through a set of detailed tasks, how the team plans to design, build, integrate, and test the selected subset of features from the product backlog during that particular sprint [10] [12] [6] [13].

Next is sprint execution, where the development team performs the tasks necessary to realize the selected features. Each day during sprint execution, the team members help manage the flow of work by conducting a synchronization, inspection, and adaptive planning activity known as the daily scrum. At the end of sprint execution the team has produced a potentially shippable product increment that represents some, but not all, of the product owner's vision [10] [12] [9] [13].

2.4 SCRUM ROLES:

2.4.1 PRODUCT OWNER:

The product owner is an stakeholder who is responsible for deciding which features and functionality to build and the order to build them [10] [12] [9] [13].

he/she must give a clear view of all the participants in the team for the goal that the team wants to reach , so he/she is responsible from the team's success in reaching the target, which is the designated product or solution appointed [11] [10].

The team rapidly builds what the product owner wants , the product owner actively collaborates with the Scrum Master and development team and must be available to answer the development team questions [10] [11] [12].

2.4.2 SCRUM MASTER:

It serves as a coach he is helping everyone to understand and participate in the values, principles and practices of Scrum and provide a leadership process for the team [10] [12] [9] [13] [10] [11] [6] [12] [6] [13].

The Scrum Master [9] helps everyone involved understand and embrace the Scrum values, principles, and practices. She acts as a coach, providing process leadership and helping the Scrum team and the rest of the organization develop their own high performance [10] [11] [12].

As a facilitator, Scrum Master is helping the team by solving the problems and remove impediments that facing the team and all that stands between the good productivity of the team he/she is also responsible for protecting the team from outside interference and takes a leadership role in removing impediments that inhibit team productivity (when the individuals themselves cannot reasonably resolve them).

However scrum Master does not have the authority to extend its control over the team, so his role is not the same as role of traditional team manager or development manager, scrum Master is not a manager, but he/she is a leader [10] [11] [12] [6].

2.4.3 DEVELOPMENT TEAM:

There are many different job titles when using traditional development approach, such as architect, programmer, tester, database administrator, UI designer, and so on. Scrum defines, cross-functional collection of these types of people who are responsible for designing, building, and testing the desired product [6] [10] [11] [12] [6] [13].

The best way to reach the goal is to make each team member doing what he/she loves so scrum team is self- organizes. The development team is typically five to nine people in size; its members must collectively have all of the skills needed to produce good quality, working software [6] [9] [10] [11] [12] [6] [13].

Of course ,you can use scrum in large projects that require more than 9 people in the development team, so you can use more than scrum called scrum of scrum each with e development team of nine or fewer people [10] [11] [12].

In traditional development approach the complete of phase or component by testing and delivery but scrum completes the sprint by performing two activities “inspect and adapt ". The first is called sprint review that means the stakeholder and scrum team inspect the solution has been built, the second is sprint retrospective which it related by continuous integration and improve the team skills. The outcome of these activities may include as part of the team's development process [10] [11] [12].

By the last two activities we complete scrum cycle and then scrum life cycle is repeats by beginning by determining the next most important feature by the product owner and do the same steps for it , after the completion of many sprints the product owner's vision will be realized and the solution can be released [10].

2.5 LITERATURE REVIEW:

With the increasing use of distributed software development, there has been growing interest in the application of different methods of software development. The software development industry is investigating the use of Agile software development methods with the distributed development instead of the traditional heavyweight methods in order to improve the development efficiency and quality [16].

Implementing agile in a distributed environment has been frowned upon because of the assumption that agile is not suited for co-located teams, mainly the problem arises from the fact that co-located teams have low level of communications but on the field another fact is the reality show by a study In the 2010 State of Agile survey conducted by, more than half the survey respondents said they are currently using Agile with both co-located and distributed teams, or planning to do so in the future.

2.5.1 DISTRIBUTED DEVELOPMENT:

The term Distributed Development was defined by Katiuscia Mannaro as "co-operation between several teams located at different sites. This includes large software companies developing a single product out of many parts where each part could be built at a separate location" [8].

Distributed development methodology became very popular, as it is results in cost saving, hiring highly qualified professionals at low costs [17] [18].

The increasingly adopted distributed development approach is cloud-base infrastructure.

2.5.2 DISTRIBUTED AGILE SOFTWARE DEVELOPMENT:

Venkatesh defined Distributed Agile Development as: "Distributed Agile, as the name implies, is a model in which projects execute an Agile Methodology with teams that are distributed across multiple geographies" [19].

2.5.2.1 BENEFITS:

Distributed development produces many benefits to both development team, and software organization. Here we describe the general benefits that can be accomplished using distributed approach.

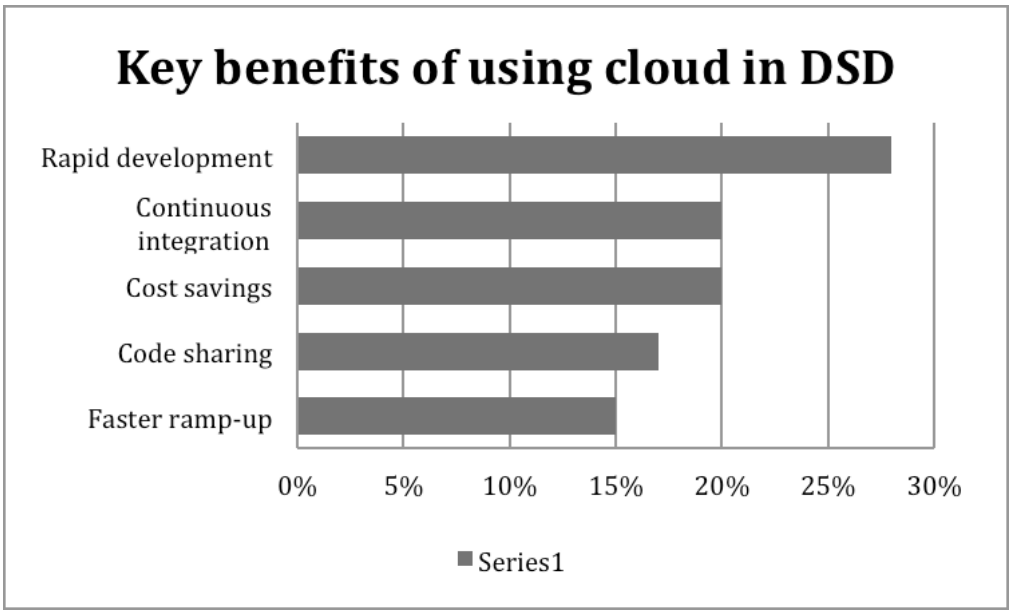


Figure 2: benefits of distributed development using cloud-base infrastructure [14]

- **Rapid development:**

A benefit that the teams gained while in the project was the ability to share cloud based tools over the distributed teams. Immediately after gaining access into the cloud, the teams were ready to contribute. A participant said "As we did not have to replicate the technical environment, we were able to start development right after the required access was available to the remote infrastructure."

- **Continuous integration:**

The centrality of cloud based software development has added the benefit of continuous integration. Cloud based software development has a unique ability which allows the developers to commit deliveries more frequently, even if other teams or units are not in control of the environment, a participant commended "We were directly accessing the remote server where the product code integration happened. We just committed all codes to the main repository."

- **Cost savings:**

Because of the high access speed to development resources and it requiring no installation pre-configured tools on local machines, major hardware and software costs have been shaved.

- **Code sharing:**

The cloud-based platform allowed code sharing across multiple teams. Although concurrent distributed programming tools were not used, the Spanish team was able to share the codebase with the Helsinki team, which had the necessary access controls.

- **Faster ramp-up:**

According to the team although several issues were encountered in the coordination and organization at the initial stages of the project, progress occurred in a more rapid pace after that. Once the environment was completely understood, the cloud based resources became very accessible and useful in manners of speeding up the software development. A participant stated said "It took two weeks to really settle on a shared understanding, but then work started,

and they (Spanish team) started to see that we could contribute quite a lot…We started to get more work then."

2.5.2.2 POTENTIAL RISKS:

Nilay Oza, and others [14] identified the potential risks of using cloud base distributed software development:

- **Dependencies:**

In terms of both technical and operational issues, created several challenges for the teams to work together. Our analysis indicated dependencies on at least two levels: operational and technical. For example, at times, one team had to wait for the other to catch up, provide feedback, or take specific actions before the development could move to the next stage.

One participant said "We had to depend quite a lot on the Spanish team to get lot of things done—sometimes we went into waiting mode or were just not able to implement things, as we had to get something done by the other team." The Spanish team member had a contradicting comment, however, "I saw them more like – 'give me work, I do my work, just my own work'.

They were depending on us for one output, one use case." One reason behind this dependency was indicated in the resource imbalance between the Spanish and Helsinki teams, as described above.

The Helsinki team worked full time for a seven week period, whereas the Spanish team worked part time for three or four hours a day.

One participant commented "We simply had more people with a full-time work commitment at our end. This meant we could do much more than the other teams thought we could. Probably, some information mismatch happened on how much work capacity was available at our end." [17]

Our results also revealed technical dependencies on the codebase as well as the overall complexity of the product's technical environment. This was reflected in the following comment: "There were many applications in the central codebase with linkages to components, etc.

It was difficult for us to determine the dependencies of these inter-connected modules, mainly because of the lack of interaction between teams." [17]

- **Unavailability:**

The new team, which did not have direct control of the cloud-based platform, were challenged by the lack of accessibility to necessary resources. The results indicated that the new team had to rely completely on the Spanish team to gain access to the cloud-specific resources if they were not previously allocated.

The general challenge that we observed was also reflected in some of the team members' comments at the Helsinki site: "The remote server is not controlled by us. If and when we lose access to it (for whatever reason), we have to contact the Spanish team, and only they can re-establish access for us."

Another comment revealed a similar challenge: "It is quite difficult to continue working when the team loses access to the cloud. Because of the common codebase and central integration, we have to wait until we are able to get access." The teams also experienced increase in uncertainties when the common cloud-based software was inaccessible.

One of the Spanish members commented "infrastructures, when fail, they generate uncertainty. Also, they have created dependency; within these limits, there was a certain dependency. [17]

For example, one day, Redmine crashed – and it becomes difficult to keep track on the user histories with the acceptation criteria." [17]

- **Code commitment and integration:**

The study showed that committing code to the proper code repository could be challenging, particularly if the cloud-based platform was not fully known to the team. Similarly, integrating code with the overall product required additional testing on the cloud platform. [17]

- **Technical debt:**

The study revealed that as multiple teams started to commit changes to the cloud-based platform, the consequent changes in other linked parts of the codebase were not visible.

More specifically, it was difficult for the new team to see the evolving impact of the changes and additions to the codebase on the cloud-based platform. One of the participants said "We did not have to worry about the platform, as it was shared by all teams, but because of frequent releases and our lack of understanding of the overall product (at least initially), we had to leave certain changes undone although we thought they would be worth implementing."

He further commented "We got specific errors in the build, and we could see that there were errors, but when we told them, they said the errors did not occur at their end". [17]

- **Additional support costs:**

The study also showed that a cloud-based platform in DSD requires additional managerial and operational support. Although the cloud-based platform has several benefits, additional overhead should be considered. [17]

2.5.3 SUGGESTIONS FOR LATER STUDIES:

Mannaro and Katiuscia [8] proposed a collection of solutions that can be used to reduce communications problems like:

- **Teleconference or daily chat:**

The team should have a tool (i.e. webcam) that can be used to make daily standup meeting.

- **Headset and microphone:**

All team members must have a headset and microphone to enable them to use computer while they are in meeting.

- **Use of synchronous meeting tools:**

Team can use any tool that provides synchronous communication feature, for example Microsoft NetMeeting, Skype.

- **Daily report:**

  They can have a feedback by reporting the work progress daily via email, either for customer or another team members.

- **Use of asynchronous tools:**

  It’s essential to have one or more wiki editor in order to manage the cooperative working.

- **Two/Three-month**ly **meeting:**

  If possible, the team must have a meeting every 2-3 months in one place.

- **Agile documentation:**

  For reason of disambiguation, all team should receive a copy of documentation.

- **Number of developers:**

  It’s recommended that number of developers should range of 10 to 15 members.

- **Using of code repositories:**

  There are many good version control like Git, CVS, which can manage code repositories, and provide useful features. Using of these tool can make development process very easy and maintainable.

  On 13 November 2013, Microsoft announced the release of Visual Studio Online, a software as a service offering of Visual Studio on Microsoft Azure platform (known as "Windows Azure" at the time). It includes online Team Foundation Server (TFS) and build-in rolling release model [1] [4] [20].

Visual Studio Customer's receives a hosted Git-compatible version control system, a load-testing service, a telemetry service and an in-browser code editor codenamed "Monaco" [3] [20]

Visual Studio online provides tools for agile teams such as [2]**:**

- Drag-and-drop backlog prioritization
- Kanban and task boards
- Sprint planning
- Bug management
- Charting & dashboards

Table 1 Summary of previous studies

STUDY	CHALLENGES	SUGGESTED SOLUTIONS
A Feature Partitioning Method For Distributed Agile	• Communication efficiency • Time zone differences • Lack of trust • Lack of control	• Formal documentation • Personal communication
Adopting Agile Methodologies In Distributed Software Development	• Communication efficiency • Common vision of the project • Time zone differences	• Teleconference or daily chat • Auricular and microphone: • Tools for the asynchronous/synchronous cooperative work in the long run • Daily Report via mails • Two/Three-Monthly Meeting

STUDY	CHALLENGES	SUGGESTED SOLUTIONS
		• Agile Documentation • Least possible Number of Developers • Project Schedules • Use of Code Repositories
Benefits Of Global Software Development Exploring The Unexplored	• Communication efficiency • Coordination and control	• Formal documentation
Scrum Practice Mitigation Of Global Software Development Coordination Challenges: A Distinctive Advantage?	• Time zone differences • Socio-cultural distances • Inconsistent work practices	• Synchronizing work hours • Synchronous communication tool-based meetings • Personal communication
Scrum Practices In Global Software Development: A Research Framework	• Communication efficiency • Coordination and control • Time zone differences • Difficult to convey vision and strategy • Socio-cultural distances	

STUDY	CHALLENGES	SUGGESTED SOLUTIONS
Challenges And Improvements In Distributed Software Development: A Systematic Review	• Group awareness. • Communication. • Software configuration management. • Knowledge management. • Coordination. • Project and process management. • Process support. • Risk management. • Quality and measurement.	• Establishment of an efficient communication mechanism • Using a version control tool in order to control conflictive situations • Application of maturity models and agile methodologies • Application of MDD approaches to automate development tasks • Systematic use of metrics tailored to the organization
Identifying Potential Risks And Benefits Of Using Cloud In Global Software Development	• Dependencies. • Unavailability of access to the cloud. • Code commitment and integration. • Technical debt. • Additional support costs.	
Practical Guide To Distributed Scrum	• Communicating with distributed team members • Time zones and working hours differences • Cultural differences • Schedule differences	• Keeping language simple • Giving everyone a chance to speak • Using group chat during meetings • Providing a translator • Working with telephones in a meeting room

STUDY	CHALLENGES	SUGGESTED SOLUTIONS
		• Use of reminders
Distributed Scrum: Agile Project Management With Outsourced Development Teams	• Difficult leveraging available resources, • Project and process management: • Lack of effective communication mechanisms. • Cultural differences • Incompatible data formats, schemas, and standards.	• Team formation • Scrum meetings • Product specifications • Configuration management • Measuring progress
The Challenges Of Applying Distributed Agile Software Development : A Systematic Review	• Lack of communication and collaboration. • Lack of management and control. • Cultural differences. • Time zone differences. • Lack of agile skills	
Successful Distributed Agile Team Working Patterns	• Cultural differences • Sharing context and priorities. • Managing customers new to agile. • Tooling for communication.	

2.6 CHAPTER SUMMARY:

Agile development is one of the most promising methods of software developments and is assured to remain in the future because of its high quality product, it’s implementation in co-located/distributed teams has become a must for all huge industries requires a method to retain the high quality of their products but also have to utilize their resources which are most commonly distributed globally.

CHAPTER THREE
TOOL AND TECHNIQUE

3.1 INTRODUCTION:

Every software product in development requires a methodology, tools and techniques to secure the success of a project, different projects have different requirements for development making you choose a set of tools and techniques each with its own pros and cons, and it's up to the development team to pick the best suiting tools, techniques and methodologies, this chapter will discuss the ones used in for the completion of this research.

As stated the selection of the wrong/bad tools, techniques and methodologies could cause project failure so the team should carefully choose the methodology that they are going to use and follow.

3.2 PROCESS MODEL:

In software engineering models and methodologies are used to orchestrate the development process of a project/product, some of these models are:

- Agile methodologies [12] [15]
- Waterfall development [12] [18]
- Prototyping [12]
- Incremental development [12] [18]
- Iterative incremental development [12]
- Spiral development [12] [7]
- Rapid application development (RAD) [12] [21]

For this project an iterative incremental model has been used to make it more agile and receptive to errors and changes, which is the most suitable model. Each model has its own advantages and disadvantages, selecting an appropriate model is critical to the development life cycle as it dictates the work flow and how you progress.

3.2.1 ITERATIVE INCREMENTAL

An iterative and incremental method is a perfect methodology for this project regarding the development team size and the project size itself, it also very easy to implement for beginners in agile and supporting the requirements of iterative and incremental development allowing developers to build the components on different increments while fine tuning each component in even iteration thus reducing the chances of error and failure with each revision assuring that all the components are ready without any hitch.

The steps of the iterative incremental model is as follows:

- Planning and requirement Gathering.
- Analysis & design.
- Implementation.
- Testing.
- Evaluation.

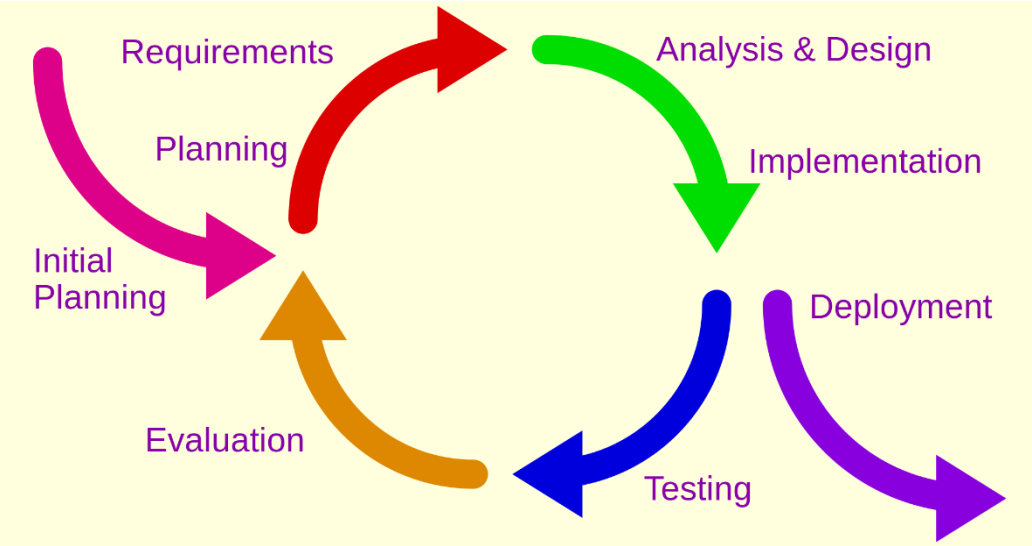


Figure 3 Iterative Incremental Model

After the basic design was finished each component of the system was taken and implemented it individually while bearing in mind that it will interface with multiple systems and that functionality could change at any moment which led to the creation of very loosely coupled components and elements, there was a

focus on providing the main functionality of each component at first then providing all the additional functions and features while enhancing all the previous work bit by bit. The tasks were distributed in the following manner, each team members chose the tasks he would like to accomplish from the agreed list of features and implemented it, and then other team member(s) would test the end result of the selected task. Once a component's main functionality is ready and usable it gets integrated into the main code. The project was incrementally built and accomplished in that manner.

3.3 TECHNIQUES:

These techniques were used to develop this research and project, there is an emphasis on selecting the techniques as their selection dictates the quality of the project. Selecting wrong or incompatible tools could increase the length of development, delay the project or even halt the progress. The techniques that are used in this project are mentioned below.

3.3.1 HYPER TEXT MARKUP LANGUAGE (HTML):

HTML stand for Hyper Text Markup Language, it's a markup language for describing web documents (web pages) structure.

Advantages:

- Highly flexible
- It supported on almost every browser
- User friendly
- Open technology
- Consistent efficient
- Easily understandable
- Designed with a feature of interaction between the web pages
- Effective.
- Provides search engine compatible pages
- Easier to maintain and update any site [22].

HTML is a must in web design, utilizing HTML to its maximum allows the creation a friendly well-constructed web page, and also HTML is mixed with other technologies such as CSS and JS to present the best looking and most fluid web page possible.

3.3.2 CASCADING STYLE SHEET (CSS):

CSS stands for Cascading Style Sheets, it's a style sheet language used to describe the look and feel for HTML documents, and it can also applied to XML documents.

HTML makes user interface for web document friendly, also it can add some animation to HTML documents [22].

Advantages:

- it separate the design form the content
- simplifies design changes
- enables you to create different style for different device
- Bandwidth Reduction [23]
- Browser Compatibility [23]
- Viewing Options [23]

CSS allows the design of more elegant web pages for users, making the site attractive towards visitors, also it helps with the separation of the document and design enabling quick changes to the design with a lot less lookup time not forgetting all the additional features it enables for a web page.

3.3.3 FONT AWESOME:

Font Awesome is a CSS/LESS icon toolkit and font created by Dave Grandy for use with the Twitter Bootstrap framework. It provide scalable, and customizable (using CSS) vector icons [24] [23]

Features

- Large icons collection in one font (more than 585 icons).
- No need for JavaScript.
- Open source under SIL OFL 1.1 license.
- Font Awesome icons are vectors, which mean they're gorgeous on high-resolution displays.
- Infinite scalability of icons.

3.3.4 JAVASCRIPT:

JavaScript is a dynamic, object-oriented, general purpose open source scripting language. It was developed at Netscape and used to make web pages interactive with the end user, it also used for validating the inputs from the user.

Advantages

- it supported on almost every browsers
- it can use for non-web based software
- it execute on the client side
- easy for learn and implements
- fast and extends functionality to web begs

JavaScript is a must for any web developer who wants an easy to use site which is a main goal, it enables many features such as auto-complete and AJAX requests such that less load time is required by the user while also increasing the dynamisms of a web page, utilizing JavaScript enables high dynamic and fluid web pages that are easy to use.

3.3.5 JQUERY:

jQuery is an open source JavaScript library, it simplifies the interaction between HTML and JavaScript, which purpose is to make it much easier to use JavaScript on your web document by writing less JavaScript code [25] [26].

Advantages:

- Fast.
- Small.
- Use CSS syntax for selection.
- Well-designed.
- Ease of use.
- Easy to extends with plug-ins.
- Great documentation.
- Cross browsers.
- It has Ajax support.

jQuery enables a lot of extra functionality with less code, enabling the development of a friendlier and more familiar environment on the platform for easier use, such as enabling the drag & drop feature with as little effort as possible, giving more time for work on other features [25] [26].

3.3.6 TWITTER BOOTSTRAP:

Twitter Bootstrap is an open source collection of tools for creating websites and web applications. It provides HTML and CSS based design templates for typography, forms, buttons, navigation, grid system, and other interface components and responsive web design, as well as optional JavaScript extensions. It is licensed under MIT License (Apache License 2.0 prior to 3.1.0) [27].

Bootstrap is compatible with all major modern browsers including Google Chrome, Firefox, Internet Explorer, Opera, and Safari browsers. Since version 3.0, Bootstrap adopted a mobile first design philosophy, emphasizing responsive design by default [28].

Using Bootstrap gird system developers can easily make a responsive web page that adjusts its layout dynamically, taking into account the characteristics of the device used (desktop, tablet, mobile phone) [28].

Advantages:

- Easy to use, just needs basic knowledge of HTML and CSS.
- Responsive CSS adjusts to phones, tablets, and desktops.
- In Bootstrap 3, mobile-first styles are part of the core framework.
- Works all modern browsers (Chrome, Firefox, Internet Explorer, Safari, and Opera) [28]

3.3.7 ACE EDITOR:

Ace is a standalone, high performance web-based code editor written in Javascript. It can be easily embedded in any web page and JavaScript application. Ace is developed as the primary editor for Cloud9 IDE and as the successor of the Mozilla Skywriter project [29].

Ace Editor boasts plenty of features which are represented in the following points:

- Syntax highlighting.
- Auto indentation.
- An optional command line.
- Work with large documents (Handles hundreds of thousands of lines without issue).
- Fully customizable key bindings including Vim and Emacs modes.
- Themes (TextMate themes can be imported).
- Search and replace with regular expressions.
- Highlight matching [30]parentheses.
- Toggle between soft tabs and real tabs.
- Displays hidden characters.
- Highlight selected word.
- Multiple cursor selection

3.3.8 CHARTJS:

Chartjs is a an easy to use customizable JavaScript library that is used to create HTML5 charts dynamically it provides multiple type of charts for the user to choose many chart options such as the shape, size , color, etc.. [31] [32]

Features:

- Easy to use.
- Dynamic.

A display of the user's burn down chart was required to increase project visibility, implementing Chartjs gives higher control over the drawn chart.

3.3.9 AWESOMPLETE:

Awesomplete is a very lightweight JavaScript library, it's used to autocomplete fields and is high customizable, it also requires no dependencies reducing the required size and lightening the load on the client side [24].

Features:

- Super lightweight
- No dependencies
- Easy to use

Auto-completion is one of the main facilities that a user might require, this is introduced in many parts of the software to increase the ease of use of the product.

3.3.10 TORNADO:

Is an open source, python, web framework, developed by FriendFeed in 2009. Tornado also has a build-in asynchronous non-blocking web server, which result of the ability to handle tens of thousands of open connections [33].

Features: [30]

- High performance web server.
- Asynchronous request handling
- Lightweight and easy to use.
- Support HTTP/s, Websocket.
- Open source under Apache license 2.0.

There are many others common used web frameworks in python such as Django and flask.

- **Flask:**

Is based on Werkzeug toolkit and Jinja2 template engine. It was developed in 2010 by Armin Ronacher. Flask intended to be micro and lightweight, so it has no components for the common functions that already exists in a third-party libraries. This results in freedom of developers to choose the libraries that fit their goals [34].

Unlike Tornado, Flask doesn't support asynchronous non-blocking requests, which result in a performance variation between the two frameworks, Andrei Fokau's comparison [35] clarify this performance variation.

- **Django**:

It follows the Model-View-Controller (MVC) architecture pattern. Django's main goal is to make it easy to build complex database driven websites, so it provides a large built-in components such as a form validation system, an internal dispatcher, serialization … etc.

Like Flask, Django is based on WSGI, an API standard for connecting Python Web frameworks to Web servers. Which is great. However, WSGI is a synchronous and blocking API.

3.3.11 PYCRYPTO:

Is a Python library that implements most of the cryptographic methods for use inside your python scripts, it includes most of the standard secure hashing functions such as SHA and RIPEMD also it includes most of the standard Encryption Algorithms such as RSA and AES.

Hashing is used in a lot of functions in the server to authenticate the integrity of requests example, an invite request should not be altered to give users the right to use join a different project than that which was assigned to it, also using cryptographic methods assures the privacy of transmitted client data [36].

3.3.12 SELENIUM:

Selenium is Software Testing Framework that is used for testing web application mostly on the client side, it provides many features such as the ability to record or playback actions on a browser, it also enables users to write testing scripts in many languages such as java and python. [37] [38]

Features:

- Multi-browser support
- Multi-language support (python, c#, etc...)
- Client side recorder
- Open source [Apache License 2.0]

Selenium has many uses it could be used for regression testing or any type of test, it has the ability to automate the web browser to act like a user without injecting any scripts, such as click or send keys to an element, this gives it the ability to provide accurate tests towards the client.

Selenium is one of the best open source testing tools maintained online it's easy to learn & use, it is used to create basic and advance tests that might encounter end users, the ability to write the testing script once and running it numerous times dramatically reduces the time and effort that is used to test out the client side [37] [38].

3.3.13 GIT:

Is an open source distributed version control system (DVCS) designed to handle everything from small to very large projects with speed and efficiency. Git was initially designed and developed by Linus Torvalds for Linux kernel development in 2005, and has since become one of the most widely adopted version control system for software development [30] [30].

Git gained a good popularity between version control systems, Eclipse Community Survey in 2014 shows that Git has gained total of 33.3% popularity between developers which means it has surpassed other leading VCS such as Subversion, the previous top VCS and Mercurial another popular VSC [37] [38].

Features:

- Open source under both GNU GPL v2, and GNU LGPL 2.1.
- Distributed, with centralized pattern support.
- Strong support for non-linear development.
- Compatibility with existing systems/protocols.
- Efficient handling of large projects.
- Cryptographic authentication of history.

3.3.14 LIBGIT2 AND PYGIT2:

Libgit2 Is a portable, pure C implementation of the Git core methods provided as a re-entrant linkable library with a solid API, allowing you to write custom Git applications in any language with bindings while maintaining the speed of the native library [39].

Pygit2 is a python binding for libgit2 to enable developers who use python to use libgit2 easily. [25]

3.3.15 REDIS:

Redis is a NoSQL open source key-value storage software that is used to store and cache key-values with optional durability, it is also referred as a data structure server because it can store multiple types of keys it supports many languages ranging from C to Perl and JavaScript [25]

Features:

- Speed.
- Persistence Toggle.
- Support for data structure.
- Hot Backup.
- Various Language Support.

What makes it special is its ability to perform thousands of request in seconds making it faster than normal SQL queries in returning requested values, adding to Redis's speed is its ability to skip writing changes to disk using volatile memory if

opted to heavily decreasing the read and write speed per request, also a main factor in its widespread is the easiness of its scalability [25].

Redis was chosen because there is a requirement for large space of non-persistent data that can be turned into persistent data when required to be facilitated for storing notifications and also the status of running virtual machine, a need to keep taps on what virtual machine is running and its options is required to give higher visibility and increase control over them [25].

3.3.16 REDIS-PY:

Redis-py is a library which enables python to communicate with a Redis Server, this tool provides python the ability to use the server without reducing the speed that is provided by implementing Redis by using a special parser developed by the creators of Redis [40] [34]. [25][26]

3.3.17 POSTGRESQL:

PostgreSQL is an object-relational database management system (ORDBMS) based on POSTGRES developed at the University of California at Berkeley Computer Science Department.

Advantages:

- Most of SQL standards supported by Postgres.
- It offers many of the advanced features like complex queries , foreign keys, triggers, updatable views, transactional integrity, multi-version concurrency control
- Users can add new data-types , functions and operators to PostgreSQL
- It available for multiple platforms [39].

PostgreSQL is supported by a large community enabling software developers to easily handle any faults and errors, also it supports concurrency also while maintaining good performance. Also it’s widely supported by community based APIs for programming languages.

3.3.18 PEEWEE:

Peewee is a simple open source ORM (Object Relational Mapping) written in python with built-in support for SQLite, MySQL and PostgreSQL [33] , it focuses on minimalism where the API is simple and the library is easy to use and understand [33].

Advantages:

- A lightweight implementation; making it easy to integrate with any web framework.
- A Django like API; making it easy-to-use.

There is another python ORM called SQLAlchemy it is an open source SQL toolkit and ORM for Python, it's characterized by flexible design that make it simple to write complex queries and use Enterprise-level APIs which in turn make code more adaptable, But it has heavyweight API leading to a long learning curve which made the choice for peewee obvious instead of it.

3.3.19 NGINX:

Nginx (pronounced "engine x") is a web server and a reverse proxy server for HTTP, HTTPS, SMTP, POP3, and IMAP protocols, as well as a load balancer and an HTTP cache. It provides high concurrency, performance and low memory usage. It was created by Igor Sysoev in 2002 and released under the terms of a BSD-like license [41].

It can ran on UNIX, Linux, BSD variants, Mac OS X, Solaris, AIX, HP-UX, and Microsoft Windows. It can be deployed to serve dynamic HTTP content on the network using FastCGI, SCGI handlers for scripts, WSGI application servers or Phusion Passenger module, and it can serve as a software load balancer [26].

It uses an asynchronous event-driven approach to handling requests, instead of the Apache HTTP Server model that defaults to a threaded or process-oriented approach, where the Event MPM is required for asynchronous processing. Nginx's

modular event-driven architecture can provide more predictable performance under high loads [26].

Features:

- Ability to handle more than 10,000 simultaneous connections with a low memory footprint (~2.5 MB per 10k inactive HTTP keep-alive connections).
- Handling of static files, index files, and auto-indexing.
- Reverse proxy with caching.
- Load balancing with in-band health checks.
- Fault tolerance.
- TLS/SSL with SNI and OCSP stapling support, via OpenSSL.
- FastCGI, SCGI, uWSGI support with caching.
- Name- and IP address-based virtual servers.
- IPv6 compatible.
- WebSockets and HTTP/1.1 Upgrade (101 Switching Protocols).
- Web page access authentication.
- Gzip compression and decompression.
- URL rewriting.
- Custom logging with on-the-fly gzip compression.
- Concurrent connection limiting.
- Request processing rate limiting.
- Bandwidth throttling.
- Server Side Includes.
- IP address-based geolocation.
- User tracking.
- WebDAV.
- XSLT data processing.

3.3.20 DOCKER:

Docker is an open source platform that allows building and running applications inside of what is called a software container, it uses resource isolation that is provided by the Linux Kernel to enable each container to run independently

but with the added overhead of starting and maintaining separate virtual machines [42] [43].

Features:

- Easy application deployment.
- Less resource usage.
- Improved performance compared to running in a hypervisor-based VM.
- Open Source [Apache License 2.0].

Docker is used to deploy apps without worrying about the environment, dependencies or conflict. It allows containers to be maintained and run in a virtualized environment to meet its every needs. Docker is required to build and run Client apps in dedicated servers in a customized sandbox that provides all the requirements of the applications while also aiding in security such as no client has more access than it's supposed to, this includes stealing other running app's data/records or infecting the host server.

3.3.21 SHELL IN A BOX:

SIAB is a free open source Ajax web terminal emulator that implements a web server that can export arbitrary command line tools to a web based terminal emulator. This emulator is accessible to any JavaScript and CSS enabled web browser and does not require any additional browser plug-ins [44].

Shell in a box is an open source web terminal emulator that uses Ajax to implement a web server that enables the export of arbitrary command lines to a web based terminal emulator. This emulator just requires JavaScript and CSS to be enabled for it to work, no additional requirements exist [44].

It was developed by Markus Gutschke, it uses Ajax to provide a look and feel of natively run shell but on a web browser [44].

Features:

- Simple.
- Useful tool to emulate a remote system's shell from anywhere in your network.
- Just needs JavaScript and CSS enabled browser.

Shell in A Box was embedded on the IDE to provide direct access to the running VMs and give full control to users [44].

3.3.22 DATATABLES:

DataTables is a free open source JQuery plugin which is highly flexible built on the bases of progressive enhancement and give the ability for advanced interaction and control over html tables [45].

Features:

- Supports almost any data source.
- Easily theme-able.
- Wide variety of extensions i.e. editor, table tools, fixed columns and more.
- Extensive options and a beautiful, expressive API.
- Fully internationalization support.
- Professional quality: backed by a suite of 2900+ unit tests.

It enables the software to provide a very easy to use product backlog for the product owner [45].

3.3.23 TOASTR:

Taostr is a JavaScript library for non-blocking notifications. It includes a JQuery Dependency with the goal of create a simple library that can be customized to the developers preference [46]

Features:

- Small.
- Easy to use.
- Expendable.

Gives a fast way to notify users of system interactions [46]

3.3.24 UML:

UML stands for Unified Modeling language it’s a general purpose modeling language for software development that intends to provide a standard to visualizing system design.

UML is used to clarify the system for the development team and as a source to return to when there any problems in understanding the overall system design. [47]

3.3.25 WEBRTC:

WebRTC is a an API definition drafted by the World Wide Web Consortium (W3C) that supports browser-to-browser applications for voice calling, video chat, and P2P file sharing without the need of either internal or external plugins. The project's goal is to enable rich, high quality, RTC applications to be developed for the browser, mobile platforms, and IoT device, and allow them all to communicate via a common set of protocols. [48]

Features:

- **Ease of use:**

  Real-time communication is supported without the need for additional applications or plug-ins.
- **Security:**

  WebRTC enforces the usage of encryption for both the media and the signaling.
- Cost savings.
- **Rich communication:**

  Enhance the communication to users and between employers with video and messaging without the need for special applications and servers.
- Un-interrupted communication.
- **Mobile Telephony:**

  By relying on WebRTC technology, service providers can enable users to access their VoIP service while on the go without specialized applications.
- **Compatibility:**

  WebRTC is supported in all major modern browsers, including Microsoft Edge, Google Chrome, Mozilla Firefox, and Opera. However as of September 2015 Internet Explorer and Safari still lack the native support of WebRTC.

The protocol is used to enable rich high quality video meeting between project members, with the lowest possible latency, because it has Peer-to-Peer architecture.

3.3.26 UBUNTU:

Is a Linux operating and distribution that is widely spread between developers and users, it's very well known for its stability and has made a name for itself in between all the other Linux distribution for what it offers. [49]

Features:

- Easy to use
- High control over OS
- Includes a lot of software packages
- High Community Support
- Tight Security

3.4 SUMMARY:

The methodology tools and techniques are an important factor in a project's success there has been an emphasis to try and handpick the best tools and methodologies that both suite the developers and the project at the same time, a lot of consideration has went into selecting everything such as Nginx's ability to increase Tornado's request handling and speed by folds rather than Tornado alone running as a server attempting to do such by its own, not to forget choosing and configuring Git because of its ability as an outstanding VCS while also paying attention to its increased popularity over other VCS.

There has been a massive to improve the software by not only going for functionality, a lot of code has been written and JavaScript libraries were searched for thoroughly to make the project a lot friendlier and less error prone, such as including auto-complete features, charts and draggable UI all in hopes of enhancing user experience.

All the tools and methodologies were carefully selected to increase the productivity of the system and improving the user experience in the platform.

CHAPTER FOUR

ANALYSIS & DESIGN

4.1 INTRODUCTION:

The Analysis phase is a crucial phase of the project any minor mistake in this part will have massive ramifications over the project, delaying or halting progress or in some extreme cases the project could be scrapped. This phase should be completed with maximum care as possible as it introduces the problems, requirements and goals of this project, its outputs are used in the product's design. The design is also very important as good design could reduce development time, resource usage and increase the overall performance of the end product and the team.

4.2 ANALYSIS:

The platform is supposed to tackle the issues described by the developers, the system has been divided into separate modules each serving a certain functionality that was required.

The system was divided into five main modules which are: IDE, Task and Project Manager, Communication and event, each to achieve its own goals and requirements.

The IDE component includes the version control, the code editor and the virtual environment which the code will be executed on. The developers provided with a method to interact with the environment to install whatever dependencies their project requires specially if they need to run their code in a specific way, The DevBox will allow them to do so.

Our version control will enable developers to have full view of their repository this also includes viewing all the branches and list of commits not to forget the team's progress in this project. It's basically a one stop shop for everything that's required code wise.

The Task and Project manager is the component that provides full visibility on all the work that is done or will be done, it includes handling all the assigned projects, the product backlog and the tasks that are included in the project. The tasks will increase the visibility of the overall project by displaying the type of tasks

whether it being a certain feature or a bug fix and showing the completion of each tasks.

The backlog will also allow the product owner to control the product backlog to support agile development.

Finally the project management segment will allow handling of the users and their roles.

The Communication is mainly the method that team members will communicate with each other, the focus is video meetings because it's the best method to convey information and also a normal chat is enabled to maximize the communication.

The software provides a video meeting to a total of 9 users which is the maximum number of a scrum team.

Our communication is based upon the light weight protocol known as "WebRTC" providing peer to peer connection with the minimum latency as possible with direct video feed from each user regardless of their location worldwide.

Event management, grants project visibility and all the upcoming dates and events that should be held, it's a must to assist in project management.

There are many problems that encounter events such as the difference in time zones, this is solved by providing event dates and time with the user's local time zone to lessen conflict and misestimating the dates.

There are multiple types of events such as meetings, "To-do" and other. They are all self-explanatory except for others as it's a place holder for an uncategorized type of events, each event will include the data of who created it, when it was created, when does it start and end and a provided description to familiarize the users with the event and what its target/goal.

4.3 REQUIREMENT GATHERING TECHNIQUES:

Requirement Gathering is a critical phase of the project, it mainly is the practice of collecting the requirements of the system from its stakeholders, and this could also be referred as requirement elicitation.

Requirements isn't just collected from stake holders, merely asking stakeholders what the requirements are rarely result in complete requirements, a professional should be able to extract all of the needed requirements from the possible sources such as stakeholders and domain experts through various methods such as interviews, observation, etc. This phase is a precursor to The Analysis phase.

A mix of structured/unstructured Interviews were used to collect requirements for our project, a team has went and met with individual from several companies that implemented agile methodologies in some of its projects.

The developers mentioned many of the problems that encountered them in an agile development environment such as code sharing, frequent build and integration that required a shared code base and frequent installation of new system dependencies due to new/changes in stakeholder requirements which caused major synchronization problems.

The interviews were very fruitful because of the experience of the stakeholders.

There were numerous meetings with Developers from Al Zarqaa, EBS and Banan-IT who gave all the information require to start working on a project of this nature and introduced all the problems and requirements as mentioned above. The questions that have been asked are appended to the end of this research.

4.4 REQUIREMENTS:

4.4.1 FUNCTIONAL REQUIREMENTS:

- **Project Management:**

  The users can create projects and users and have a privileged control over them.

- **Task Management:**

  The users shall be able to view, add and edit all the tasks that belong to a certain project and this component should project enhance project visibility.

- **Better Communication:**

  Team members should be able to have a video meetings and chat using our platform.

- **Code Editing:**

  The users should be able to view and edit the project code from the platform.

- **VCS**:

  The users will have access to all versions of their code via a deployed VCS on the platform.

- **Code Execution:**

  The users can run their code anywhere anytime they like regardless of their machine specification.

- **Event Management:**

  Users should be able to view, add, edit and delete project related events.

4.4.2 NON-FUNCTIONAL REQUIREMENTS:

- **Portability:**

  The users shall be able to use the platform from anywhere.

- **Security:**

  Many levels of security were put into mind when designing the software such as running code will be ran in an isolated sandbox environment so that the server will not be effected by any malicious code, and no application will be able to embezzle data and information from another. In the other hand, the system must has an excellent access management to company's projects and customers, which protect user privacy.

- **Usability:**

  Users should have no trouble using the platform with maximum ease of use as possible also a very fast learning curve.

- **Availability:**

  The service is to be available to all users at any time.

4.5 SYSTEM MODELS:

4.5.1 USE CASE

The Following diagram describes the main functionality of our system

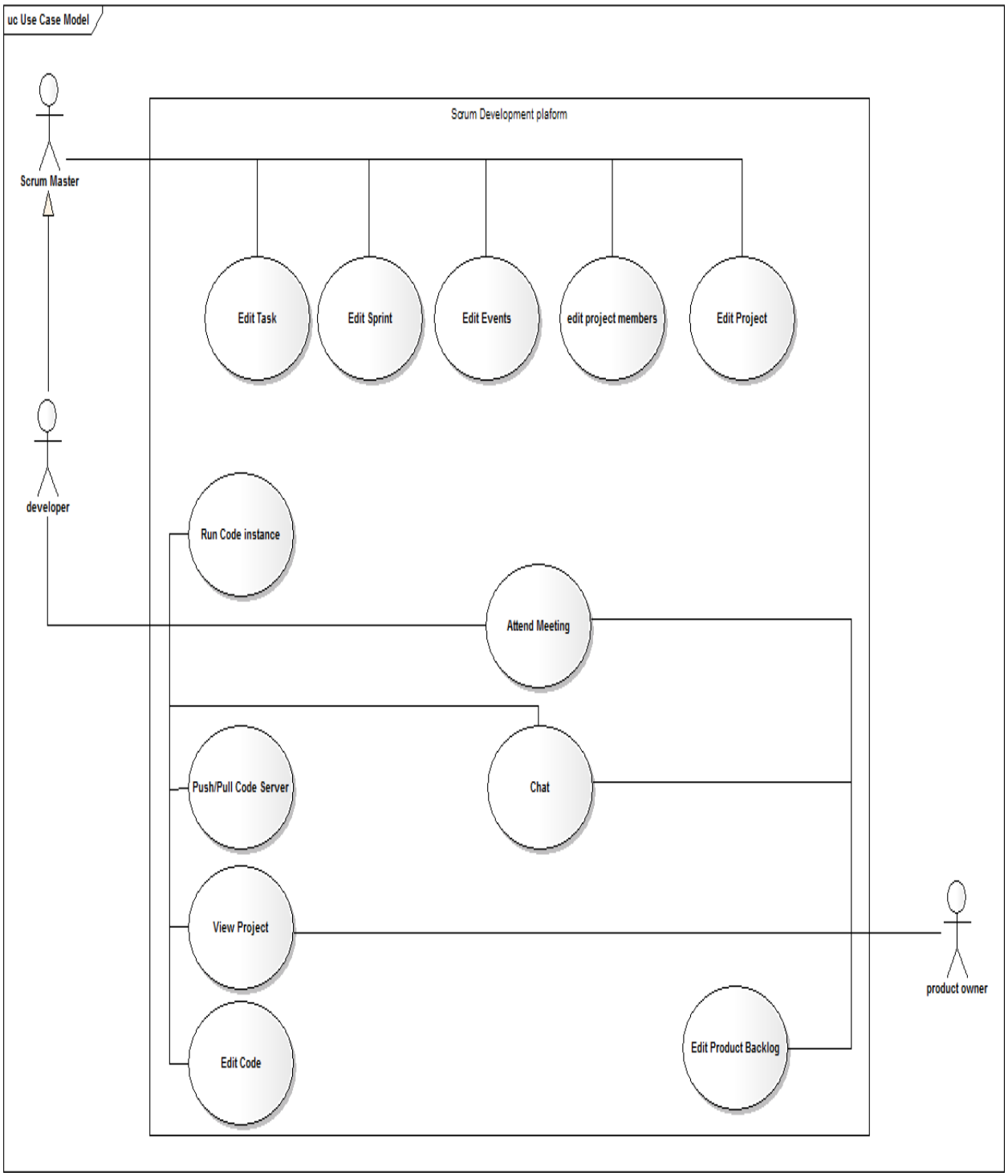


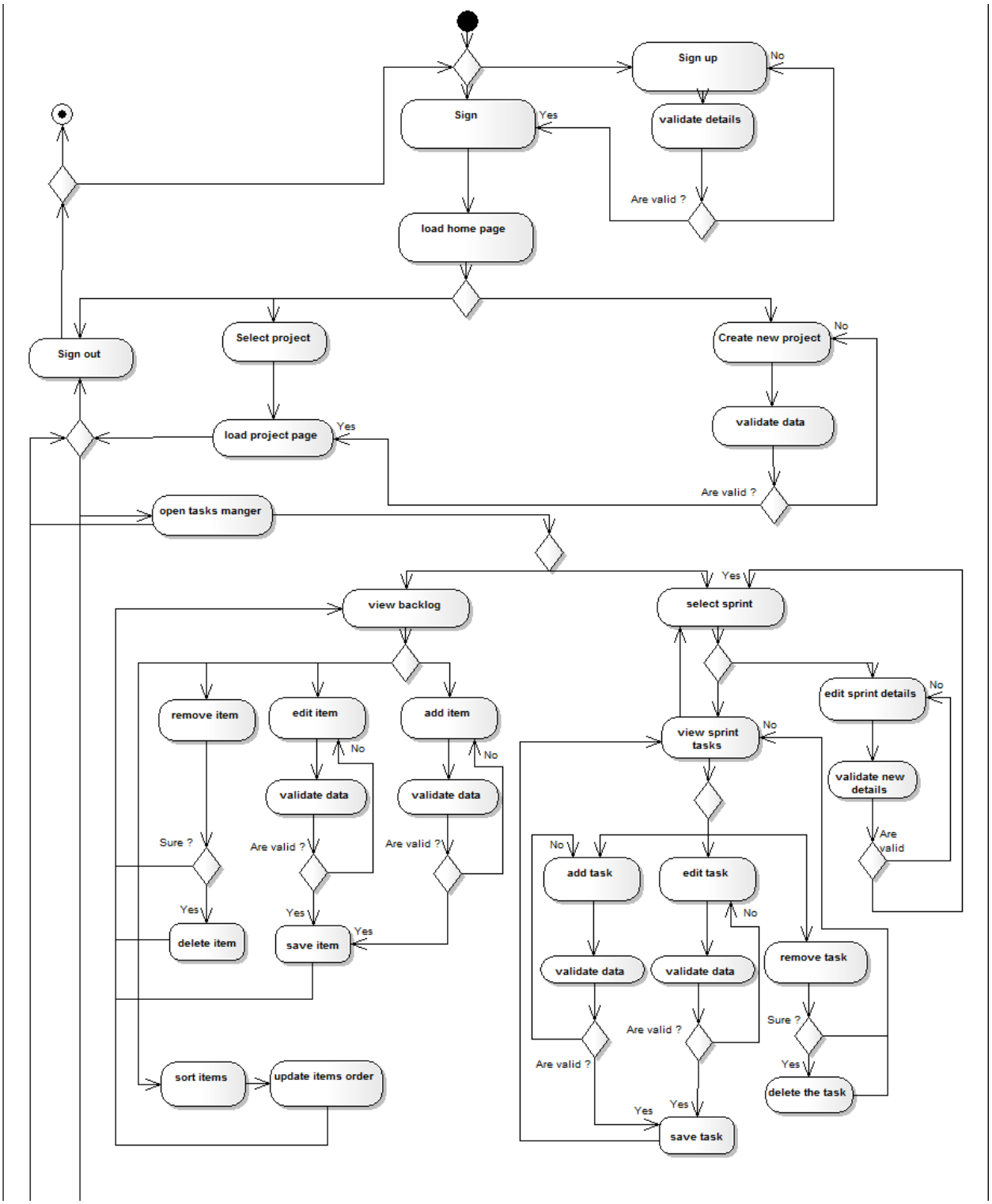
Figure 4 System Use Case Model

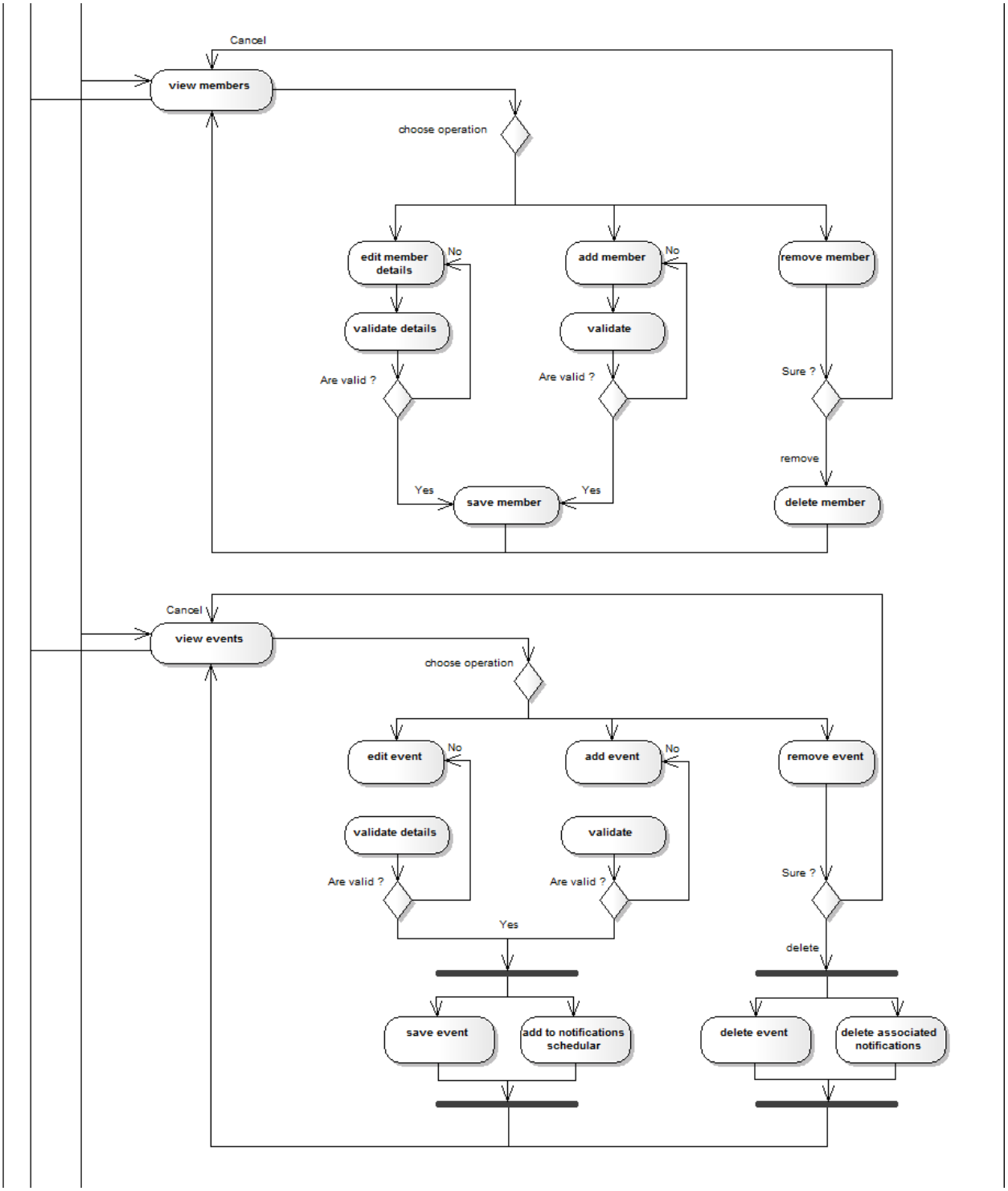
Table 2 System Use Case Description

Use Case Name	**Use Case Description**
Edit Task	The user will be able to add and remove project tasks and also change the information of existing tasks.
Edit Sprint	The user will be able to introduce new sprints into the project and remove them and change the details of existing sprints.
Edit Events	The user will be able to add and remove events and make changes to their details.
Edit Project Member	The user will be able to add new project members, remove old ones and change the roles of existing members.
Edit Project	The user will be able to add new projects, delete old ones and edit most of the project info.
View Project	The user will have the ability to view some of the project information such as the Backlog.
Edit Product Backlog	The Product owner will be able add, edit and delete the desired project features
Chat	The user will be able to have a text conversation with a specific set of members (project members).
Attend Meeting	The user will have access to a video meeting with other project members.
Edit Code	The Developer will be able to view project files, add new ones, remove undesired files and edit the files of the project.
Push/Pull Code	The Developer can remotely pull/push code into the server without using the platform via a dedicated Git server.
Run Code Instance	The Developer will be able to execute the project on a dedicated remote server

4.5.2 ACTIVITY DIAGRAM

The following diagram displays all of the activities of the project that are deduced from the analysis phase and how they will be executed if a user has access to the components and features, noting that the system will validate if a user has the right access level to a certain string of activities or not.




Cancel
view members
choose operation
edit member details
No
validate details
Are valid ?
add member
No
validate
Are valid ?
remove member
Sure ?
remove
Yes
save member
Yes
delete member
Cancel
view events
choose operation
edit event
No
validate details
Are valid ?
add event
No
validate
Are valid ?
remove event
Sure ?
delete
Yes
save event
add to notifications schedular
delete event
delete associated notifications

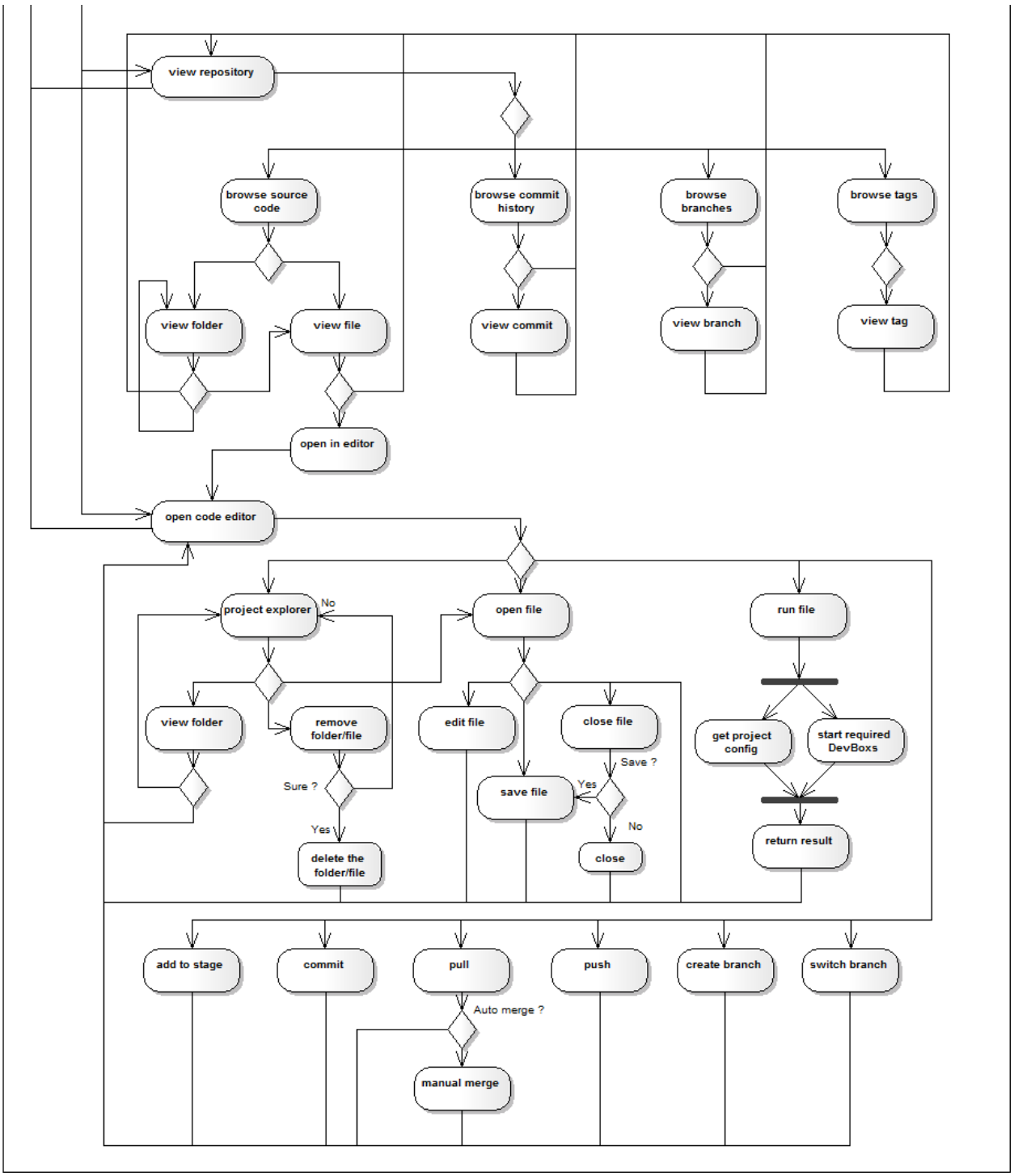


Figure 5 System Activity Diagram

4.5.3 CLASS DIAGRAM

This Diagram describes the entire system and the interaction between classes.

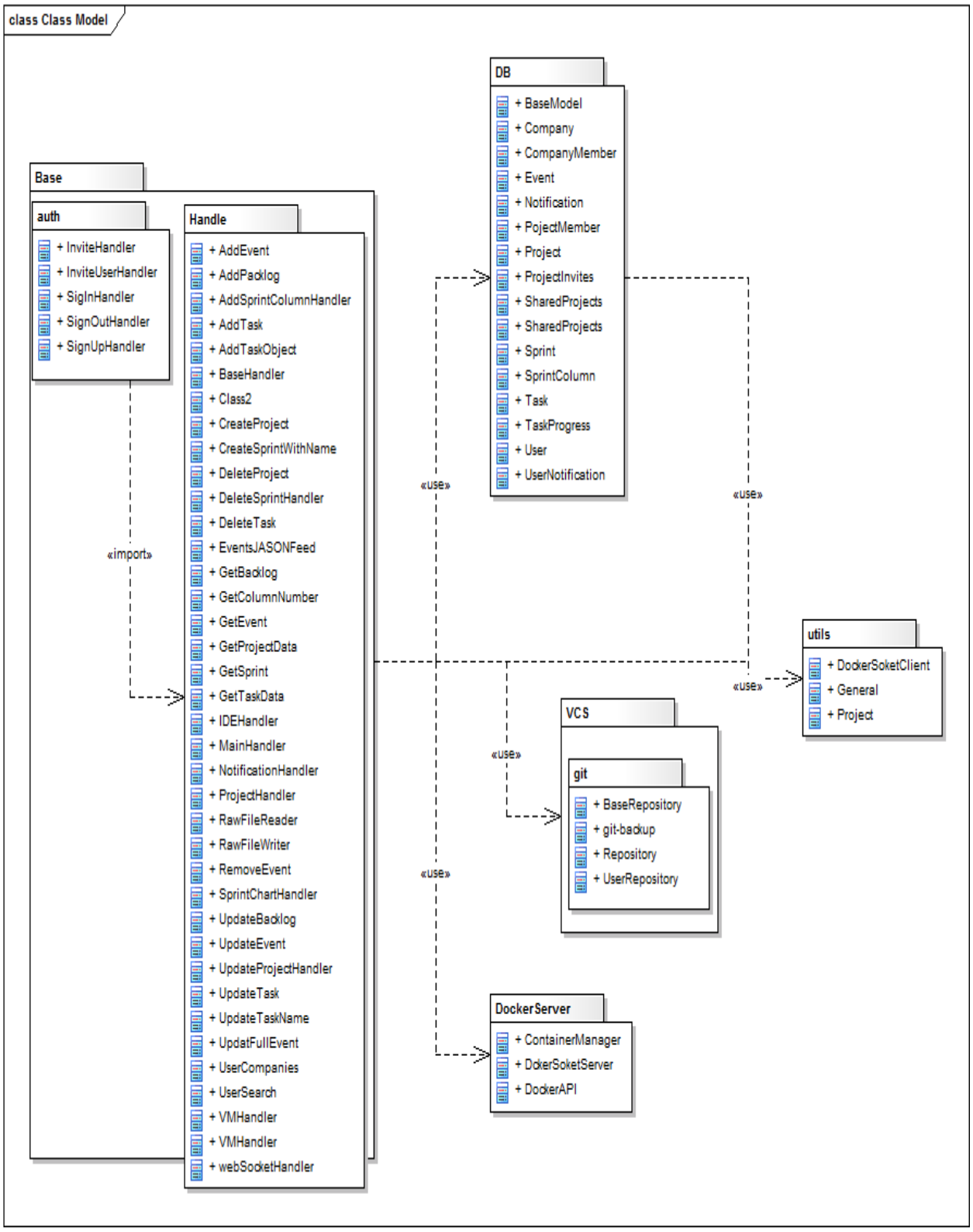


Figure 6 System Package Diagram

The Handle Class is responsible for receiving, interpreting and handling all the user requests whether it being a page that is supposed to be rendered or a functional user requests ex(update task).

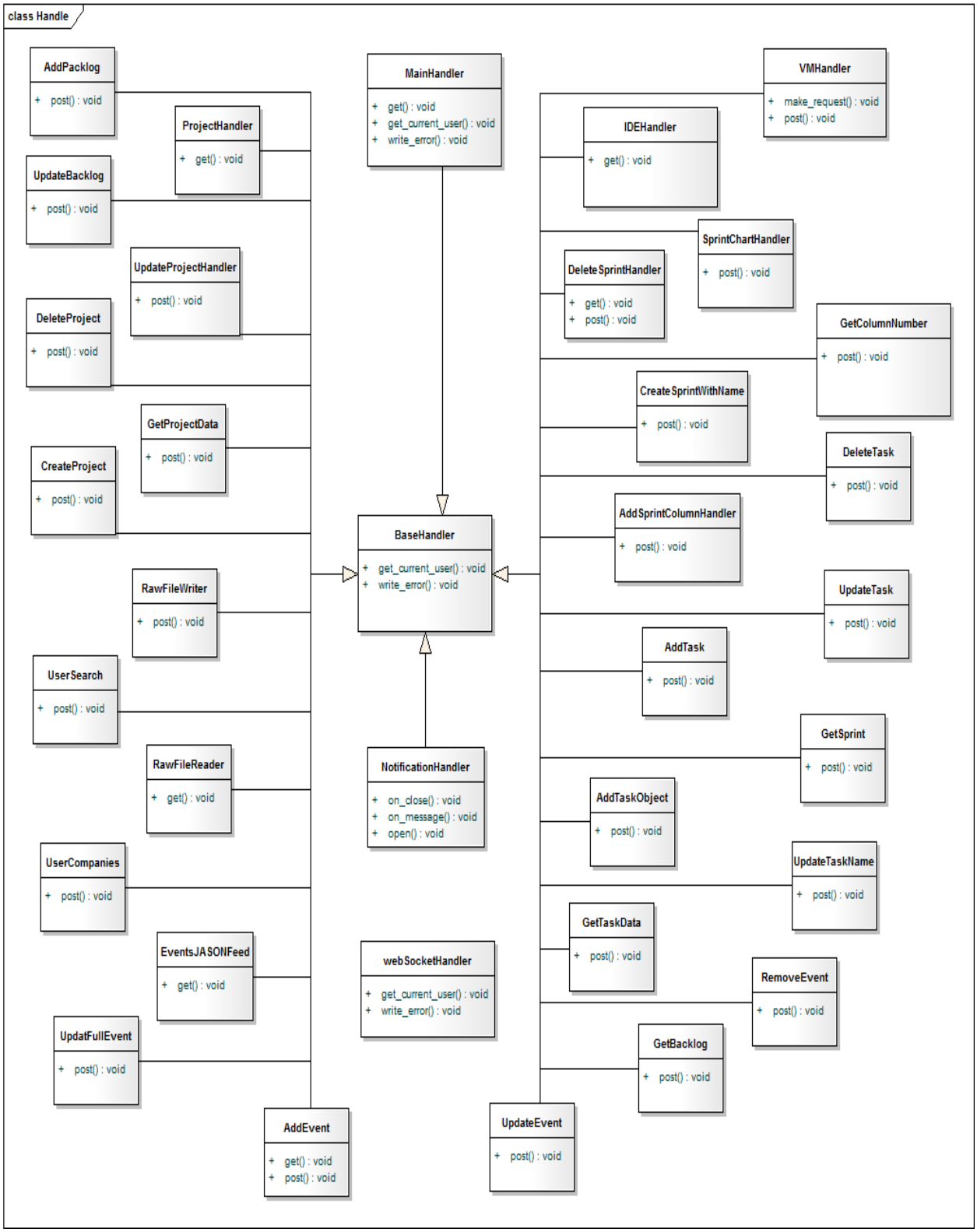
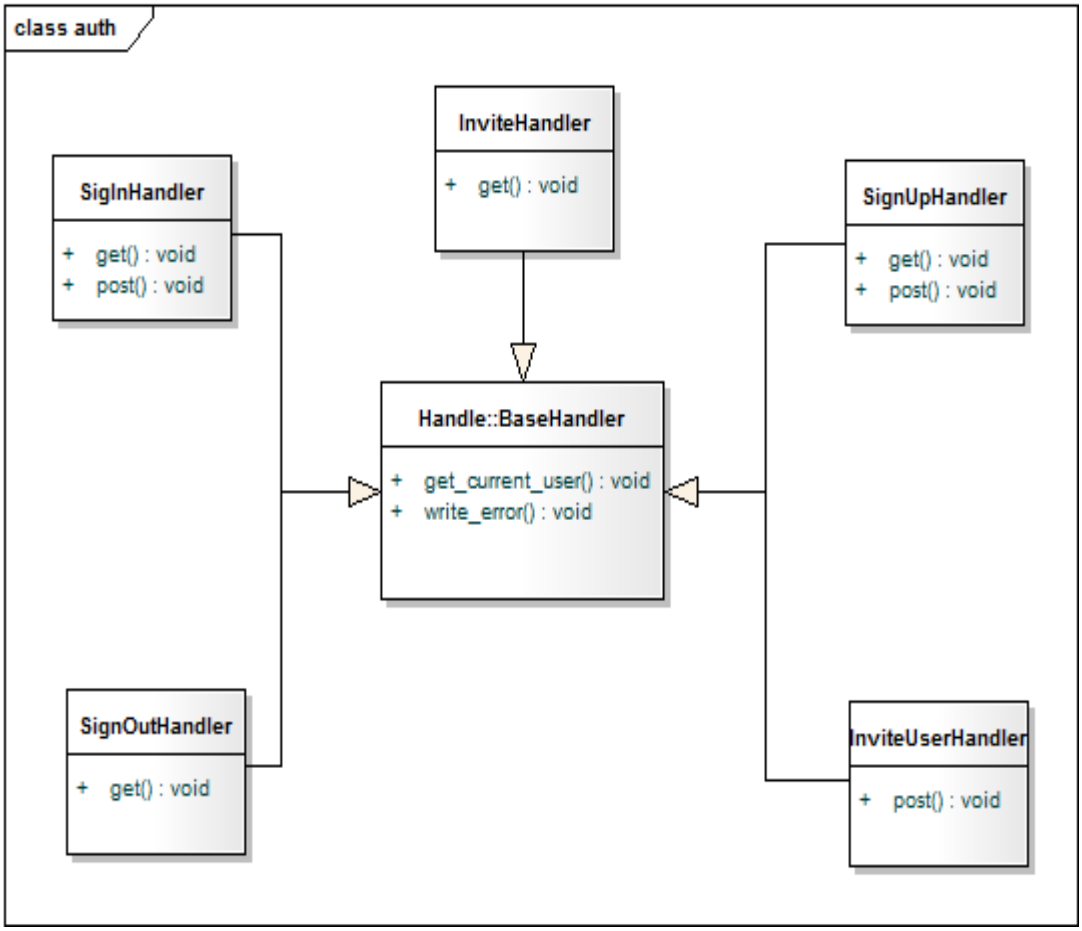


Figure 7 Project Handlers Class Diagram

The Auth class's main focus is to authenticate users and authenticate unregistered user's project invites, also an additional functionality is to register new uninvited users as the code already exists in it.



Authentication Class Diagram8 Figure

The Utillity class are commonly used functions between classes and some communication methods.

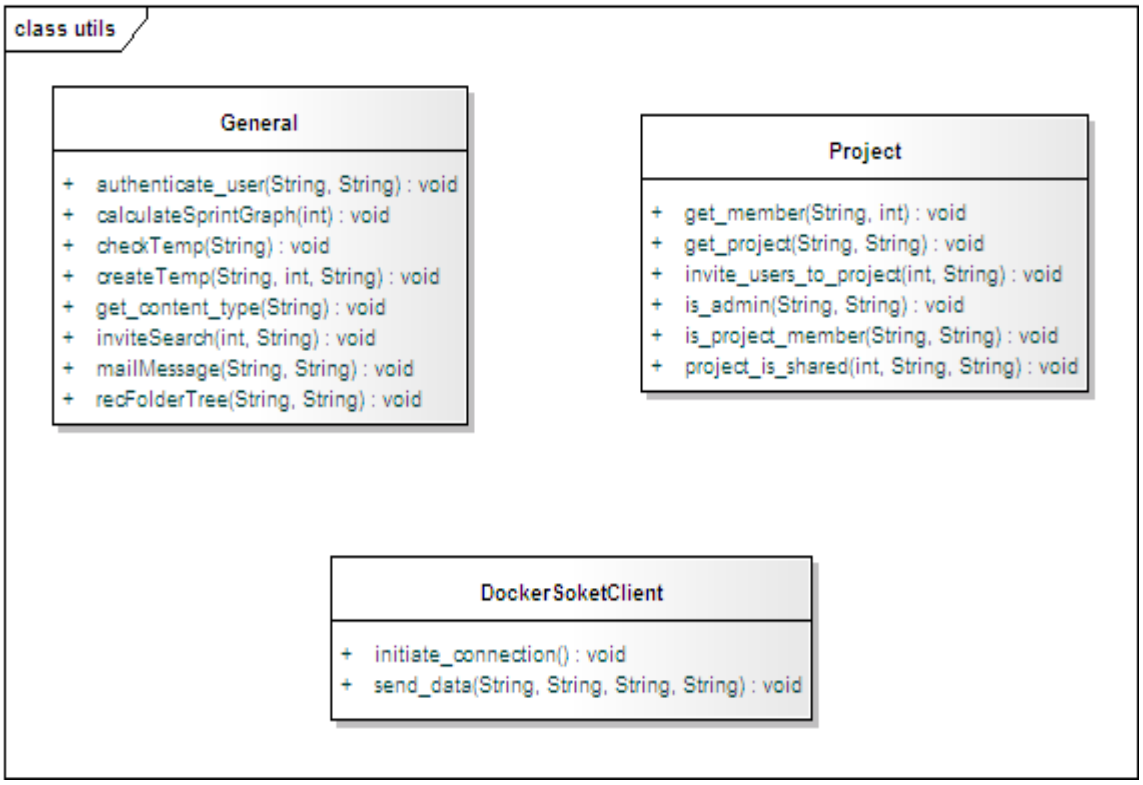


Figure 9 Util Class Model

The DockerServer class is responsible for instantiating virtual machines correctly upon request by parsing requests and turning them into linux shell scripts.

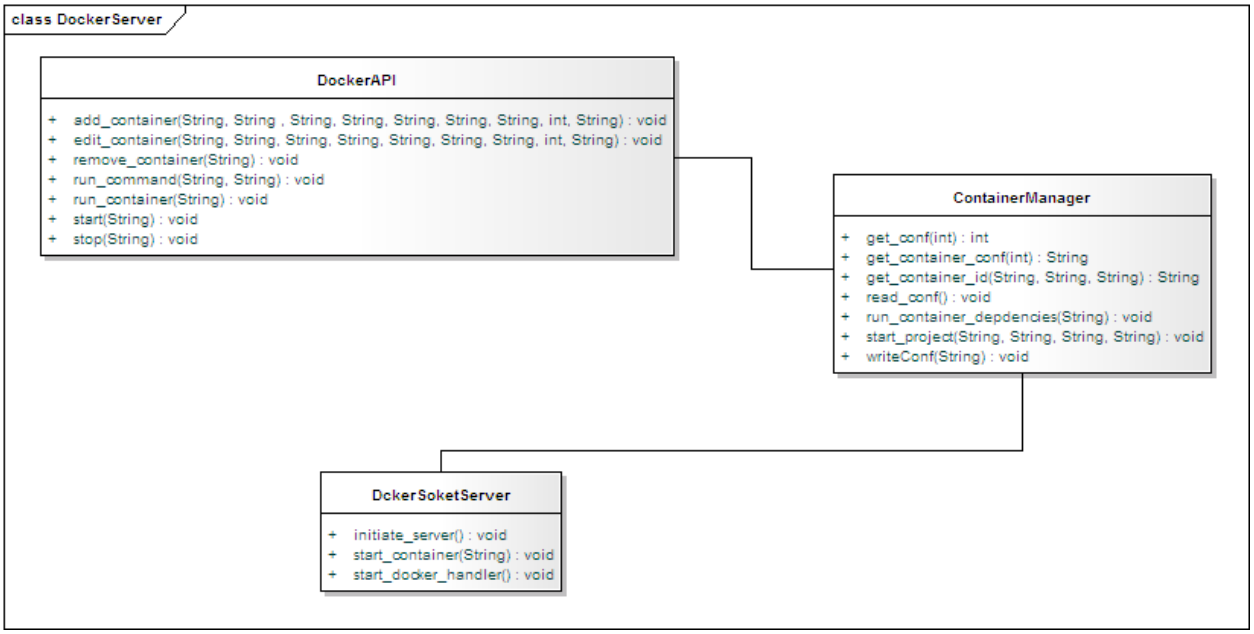


Figure 10 Docker Server Class Model

The Git2 class is the class that deals with handling git request such as creating user repositories, checking them, pulling and pushing them out, all through the use of python methods for simple usage.

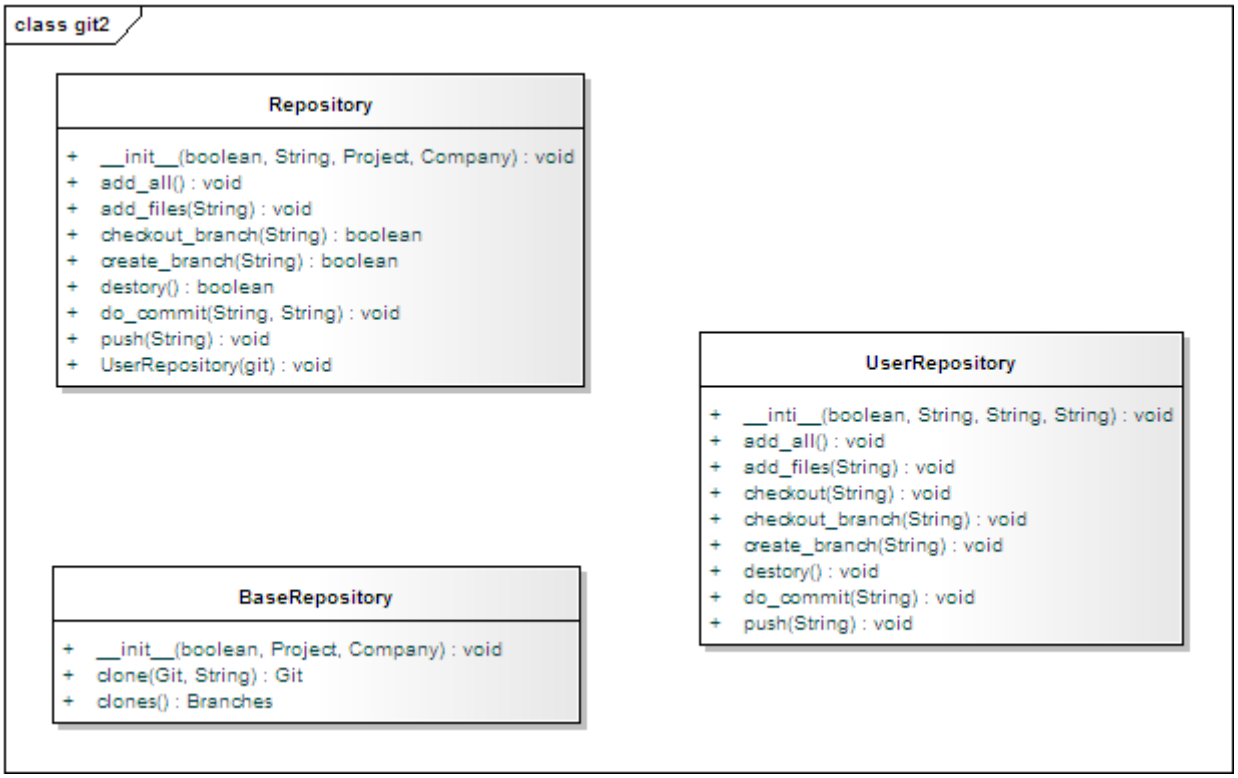


Figure 11 git2 Class Model

4.5.4 DEPLOYMENT DIAGRAM

The following figure describes the physical deployment of the system and it communicates.

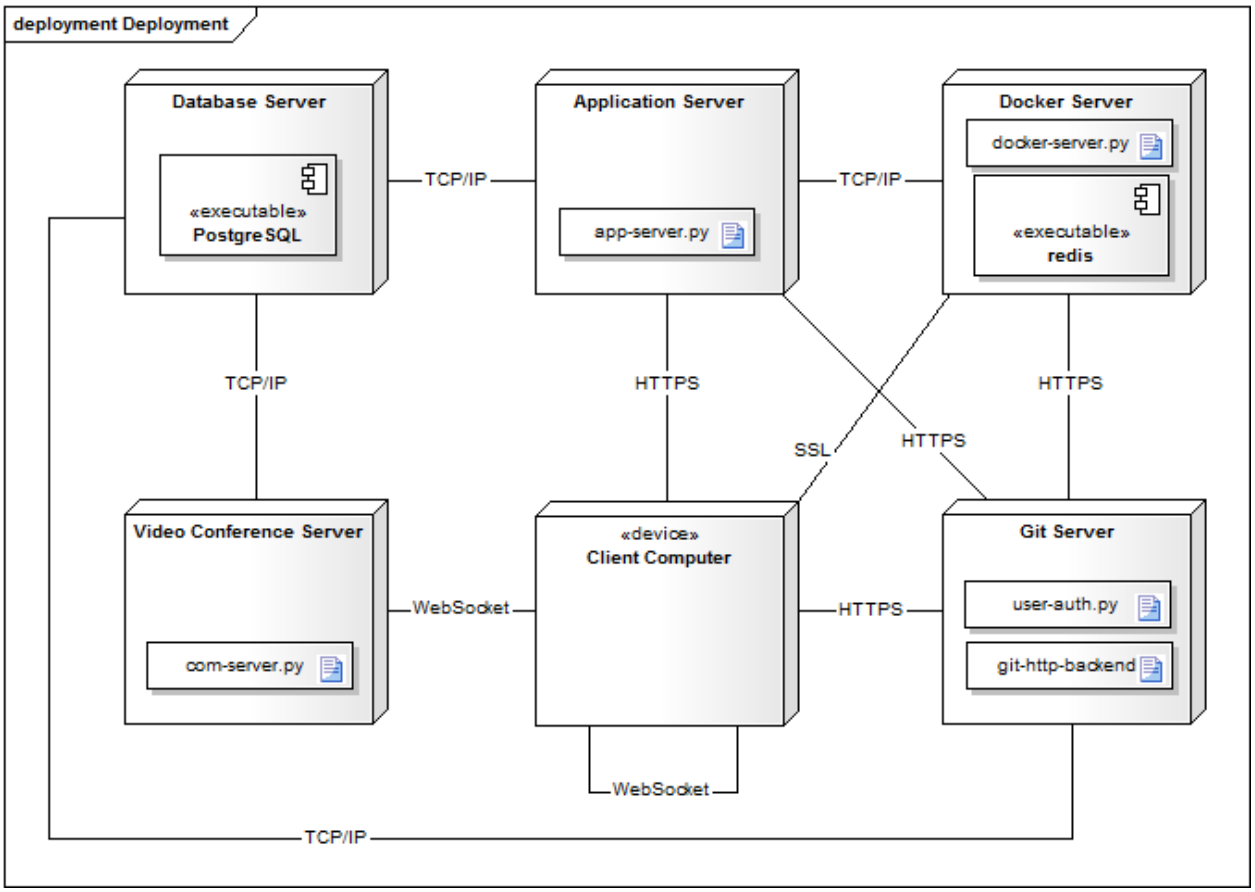


Figure 12 Util Class Model

Table 3 System deployment Description

Artifact	**Description**
app-server.py	This is the main application which will run and handle off the user request
docker-server.py	This server will facilitate the use of docker throughout TCP requests
com-server.py	Allow users to communicate with each other
user-auth.py	Authenticate all requests that are received to git through https and validates them
git-http-backend	Runs the Git backend server

CHAPTER FIVE
IMPLEMENTATION

5.1 INTRODUCTION:

This chapter shows the implementation steps of this software and some of the testing that has been done.

5.2 IMPLEMENTATION STEPS:

There are multiple servers at work for the software to be completely functional. For the IDE to function.

5.3 HOW SYSTEM WORK:

This is the login screen which is used to login to software

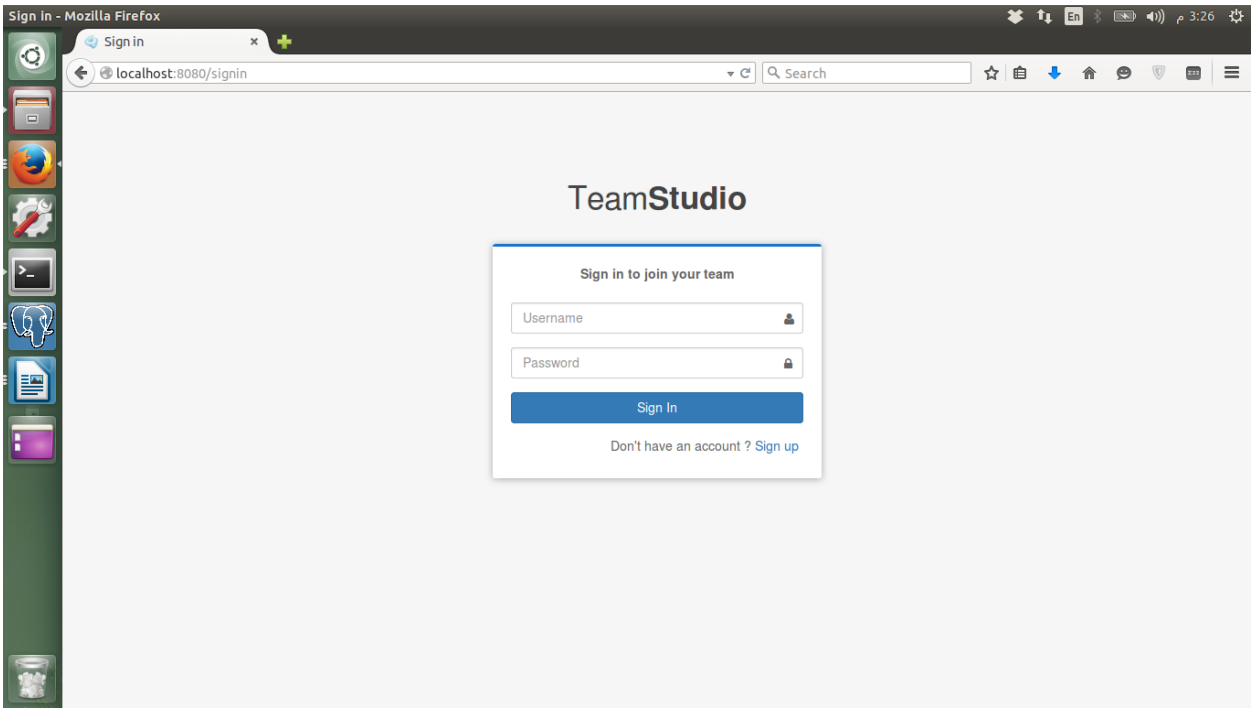


Figure 13 Login Page

The sign up Page where the users registers his account in the service

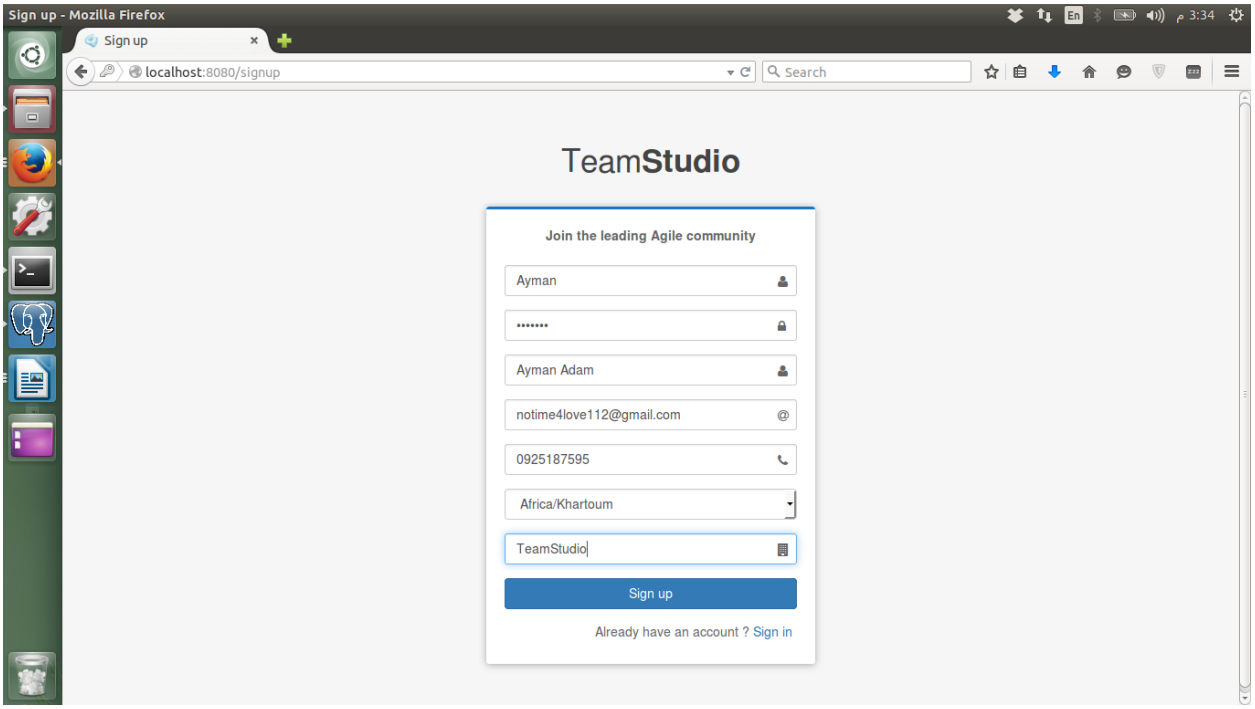


Figure 14 Sign up

When a user is invited to a project and he's not registered in a site, an email will be sent to him where he can register with a lock on his email and he'll be directly in the project group when he logs in.

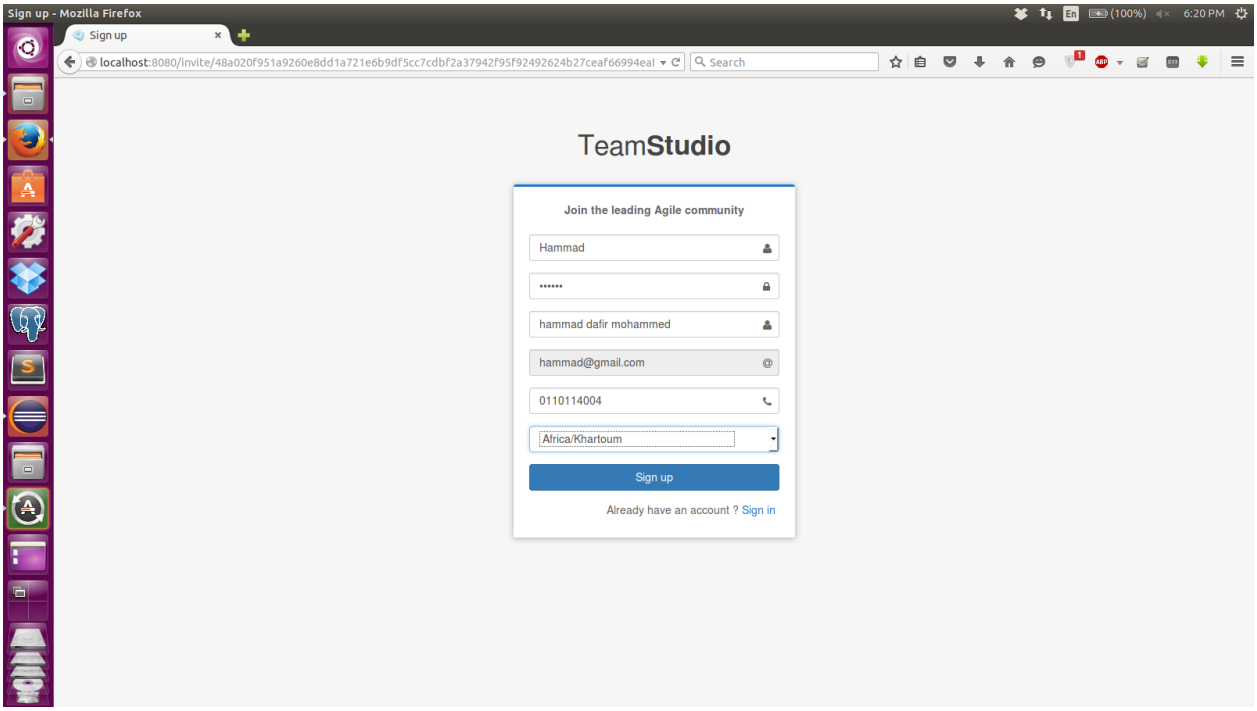


Figure 15 invite

The home page is the project page, where the user can view all of his project, if the users wants to logout at any time the button is located above.

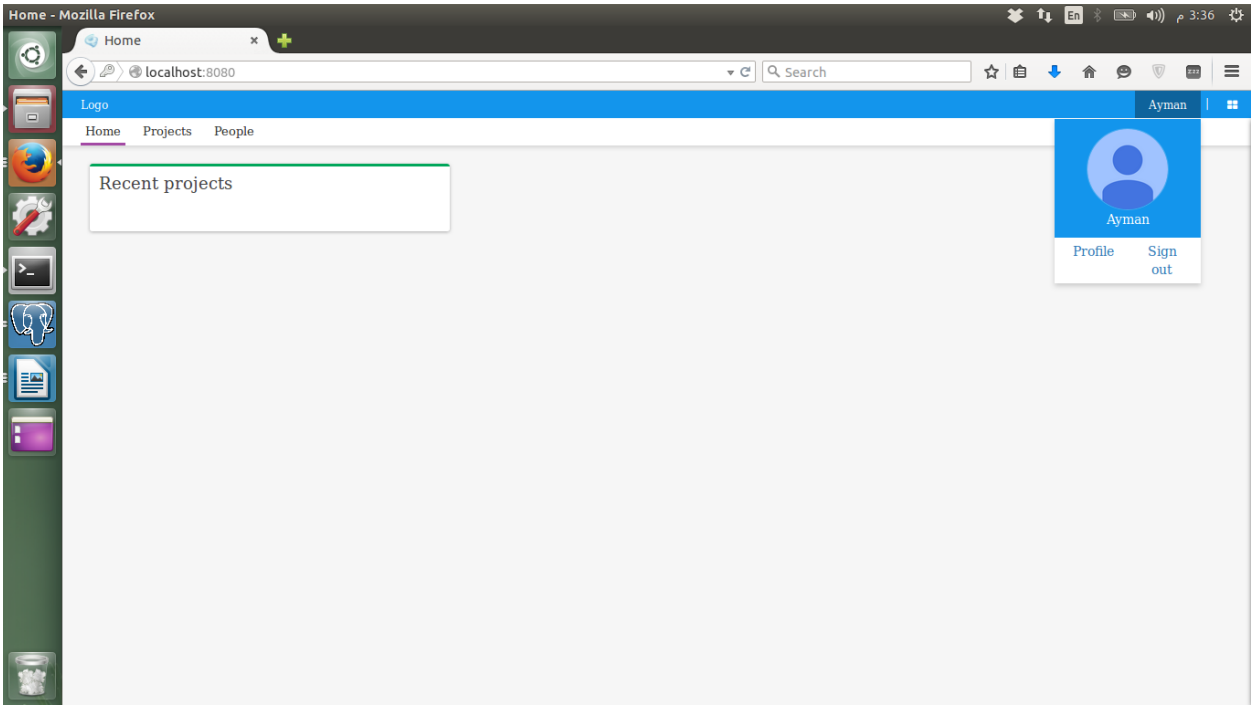


Figure 16 user logged in

A user can create a project from the project tab by pressing the [+] button in projects, after that the user will be prompted to enter project details.

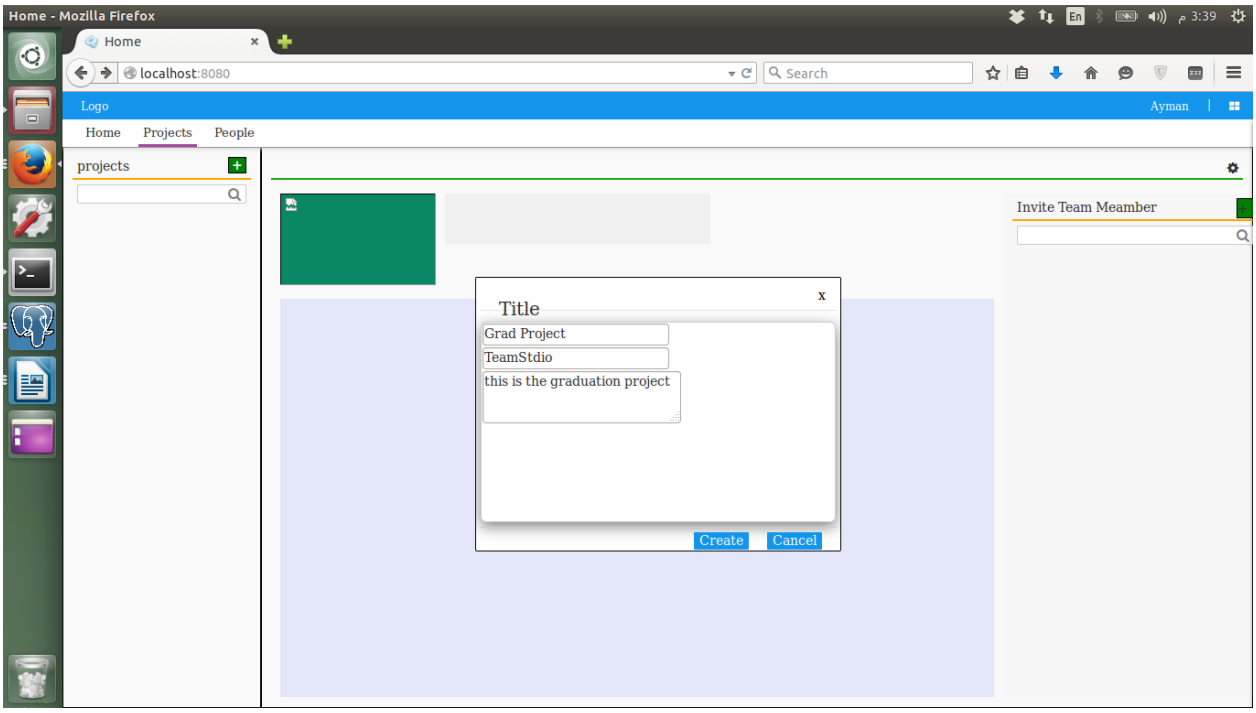


Figure 17 add a new project

A user can click on a project to view it and its details, on the right side are the members of the project, also the user can invite another project member if he has the right access level by press the [+] in invite Team Member.

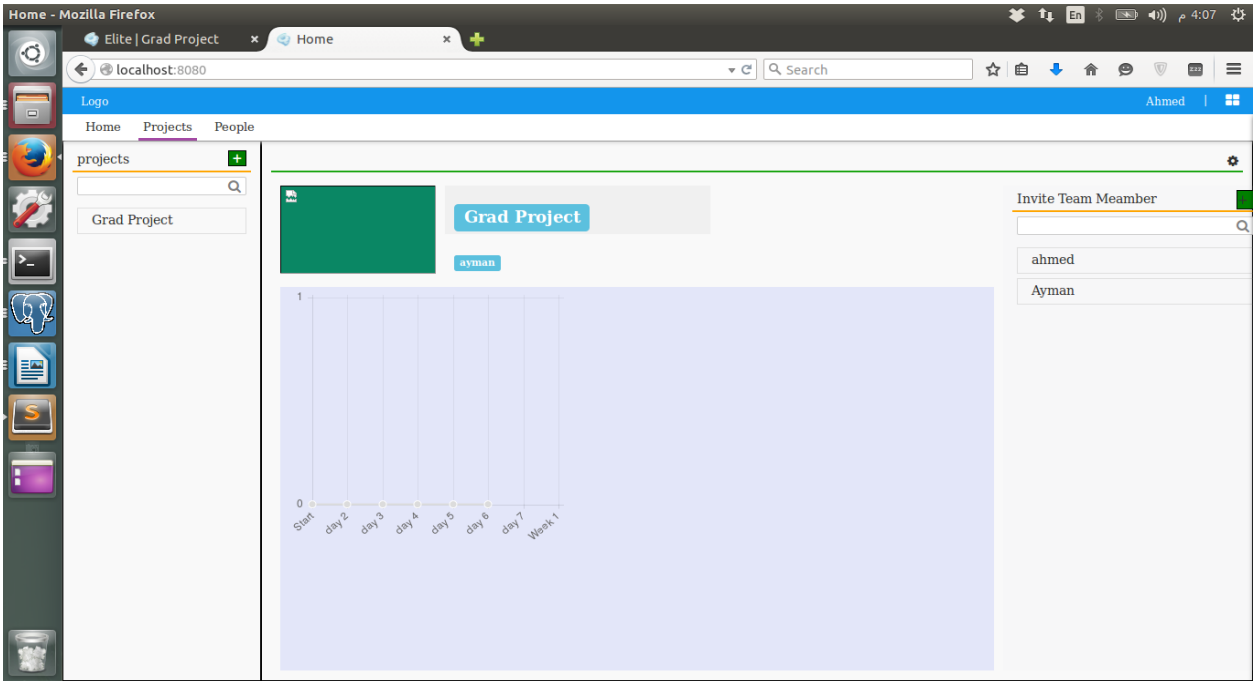


Figure 18 project page

These are the recent projects that the user has used, if users don’t want to go and browse all of their projects, in this tab he can view his most recent projects which are ordered by last access and have entry to them.

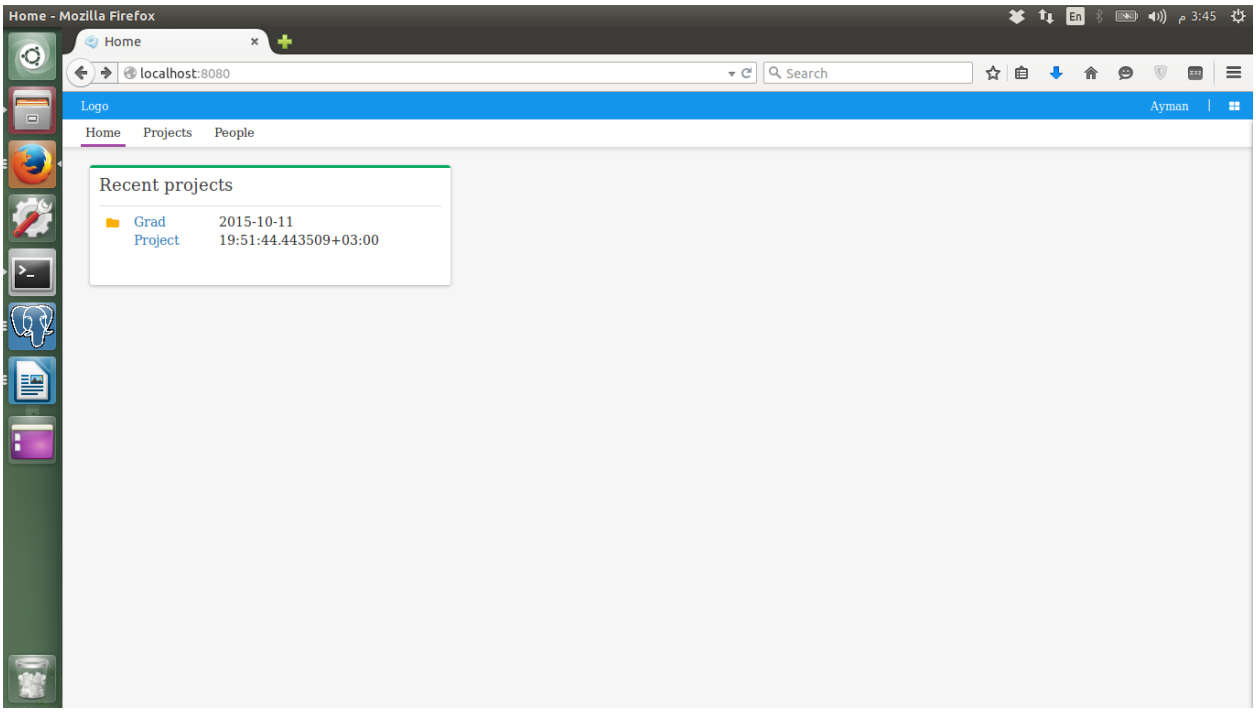


Figure 19 home page

This is a project page when you press on a project, this page will open and show everything that a user could view such as the backlog, the code and the work progress.

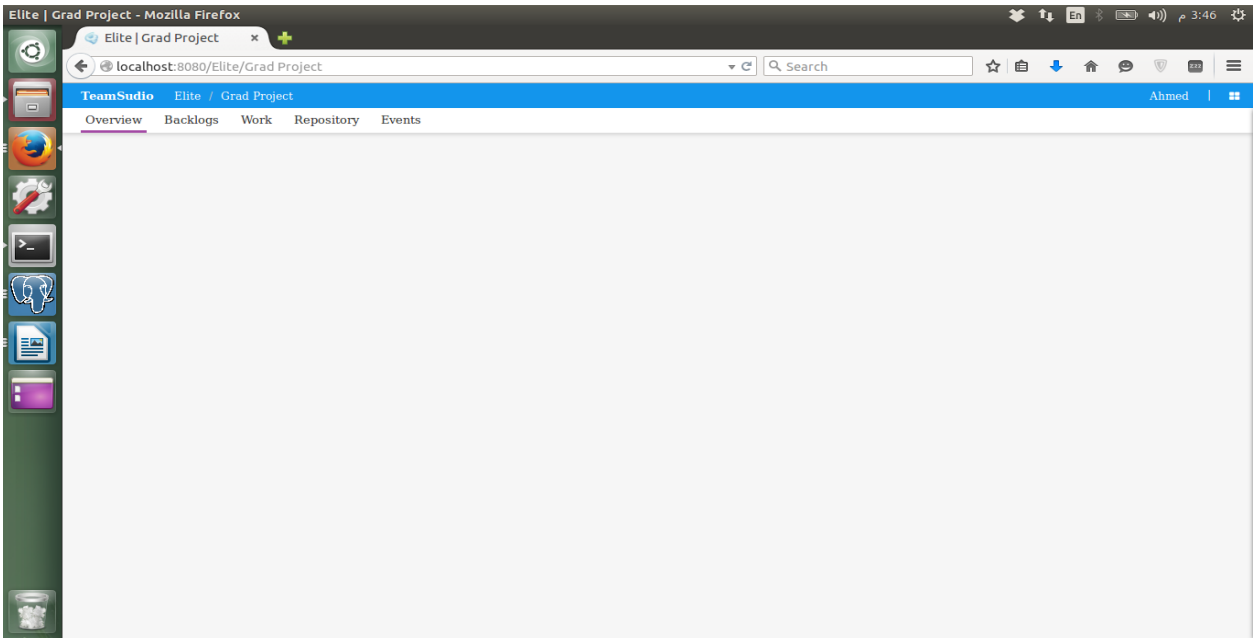


Figure 20 project home page

This is the product backlog tab, the product owner here can create new, delete and edit backlog items.

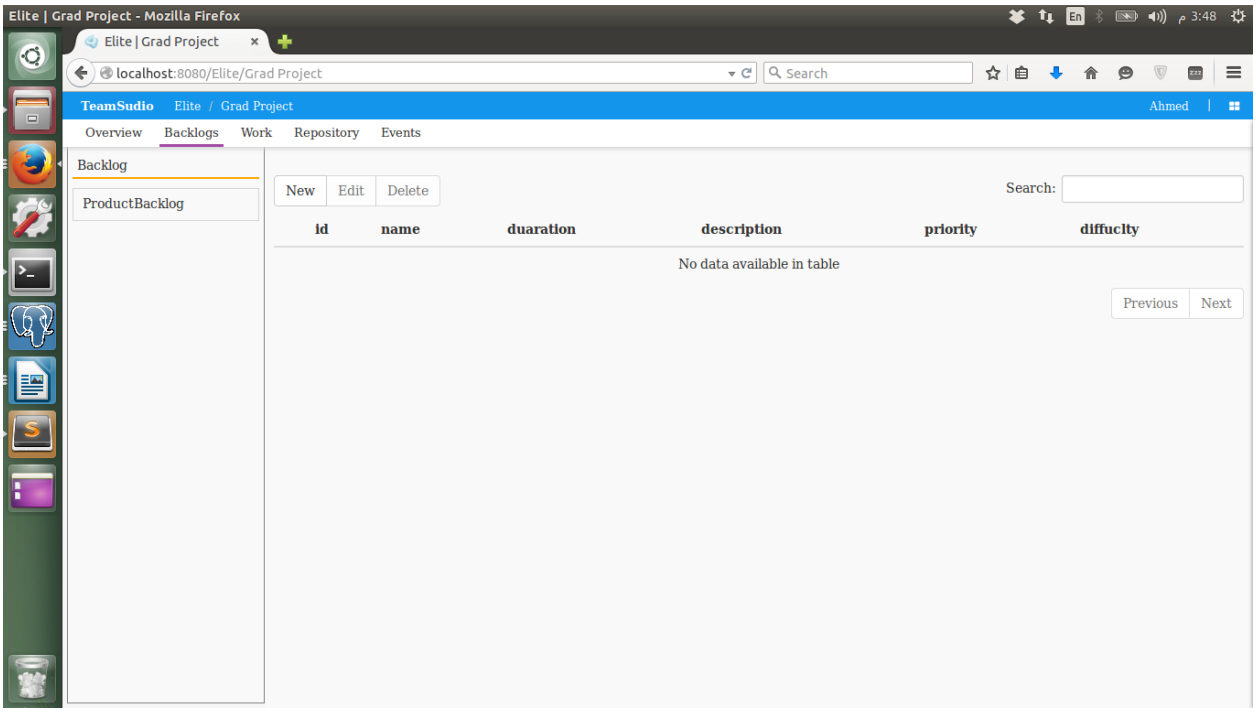


Figure 21 product backlog view

The sprint tab is the place where all the task management happens, a user can create new sprints here if he has the correct access level.

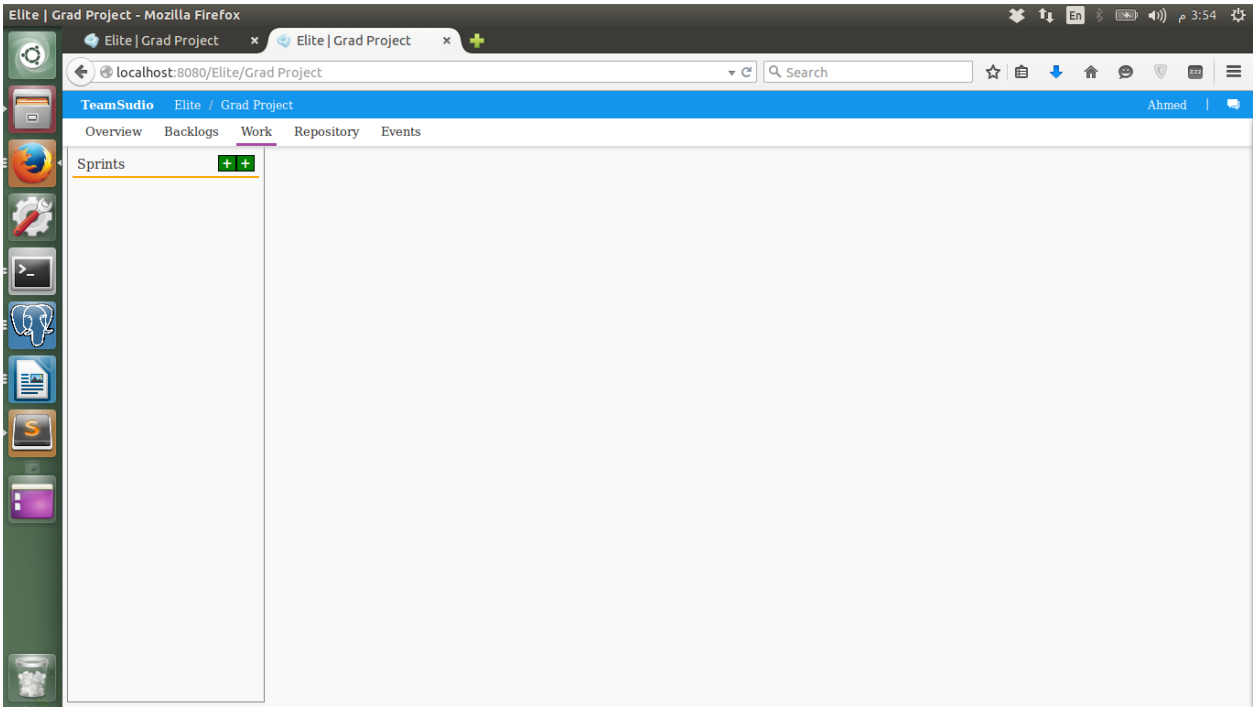


Figure 22 view sprints

When the green [+] is pressed a new sprint will be created for that project.

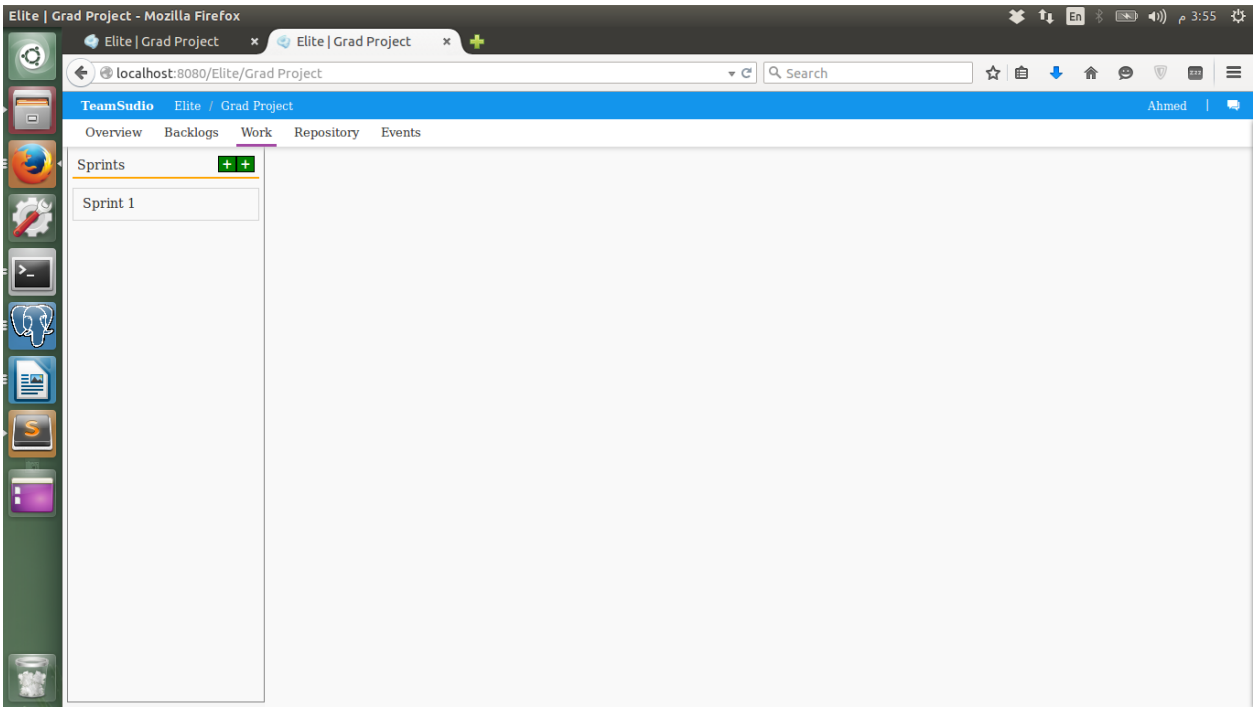


Figure 23 sprint added

The sprint data appears when a user clicks on a certain sprint, this will display all the tasks in a sprint and the phases they are in, the user can create a new task by pressing the [+] button which is placed in the New column.

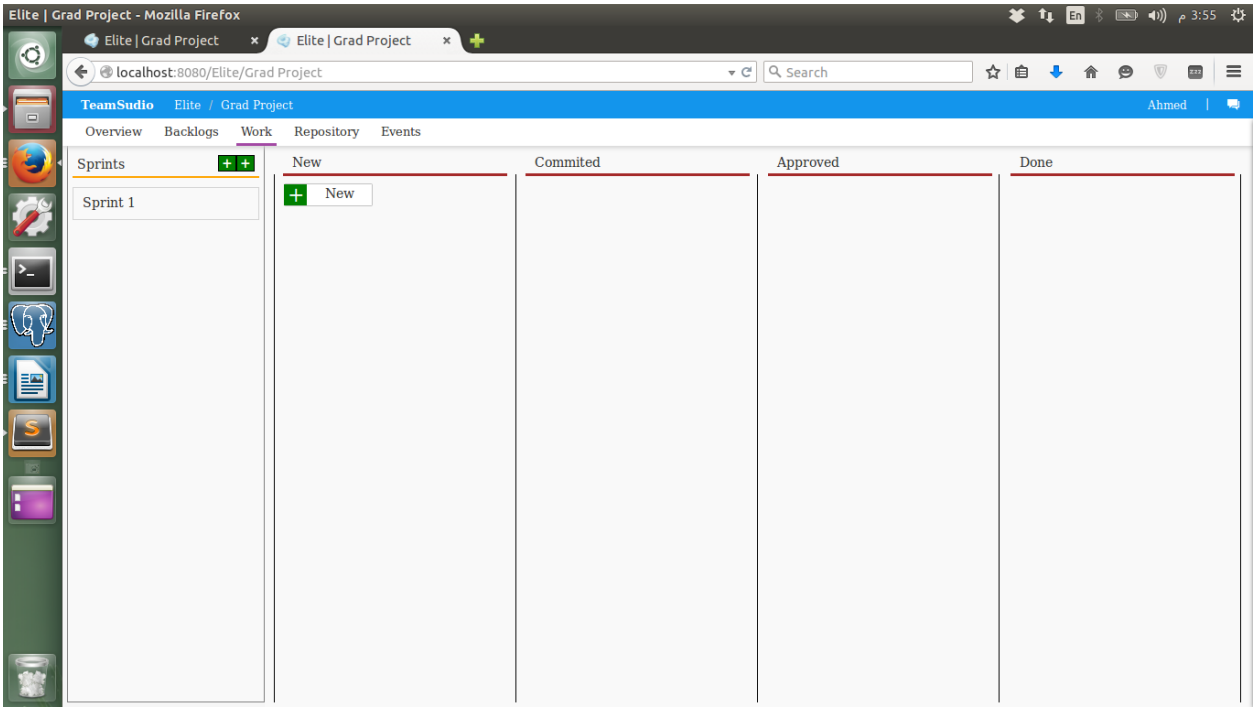


Figure 24 sprint loaded

Here when a user clicks on new task a new tasks is created with minimal data for quick addition of tasks which can be edited later

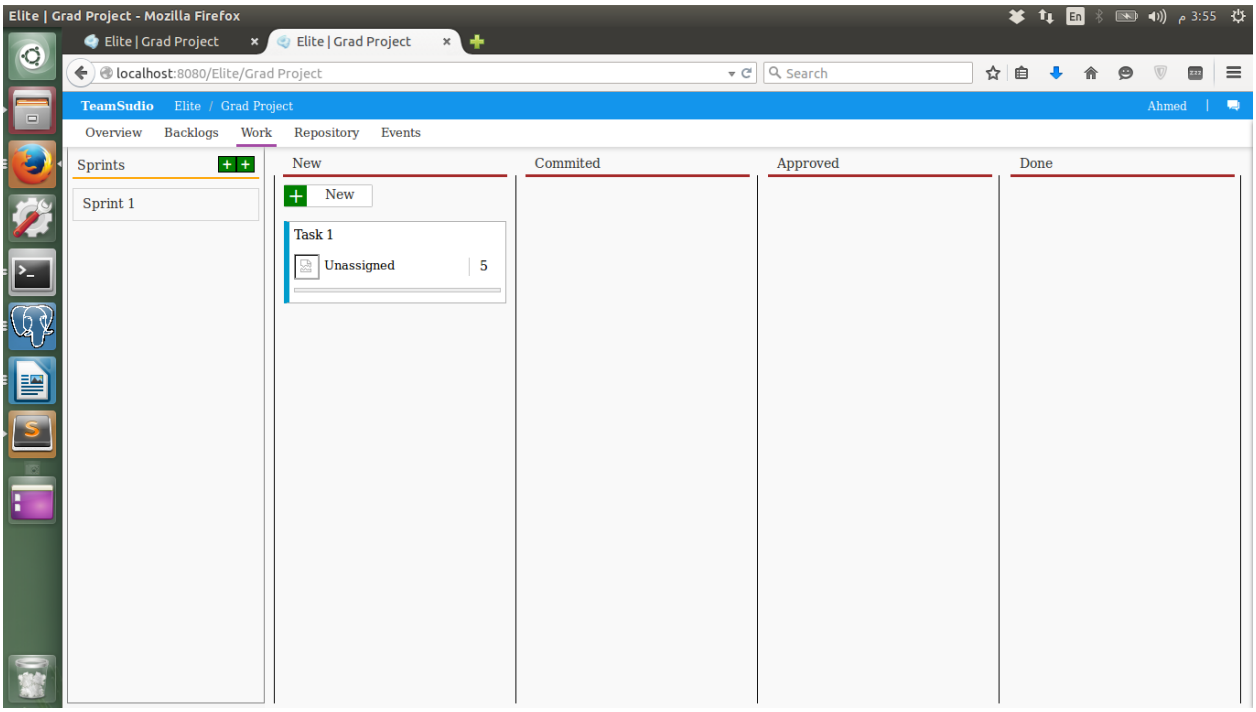


Figure 25 task created

The user can move tasks between windows by just a small drag and drop action, such as if the user would like to change Task1 from being newly created to being approved or committed he just needs to drag them into the required columns

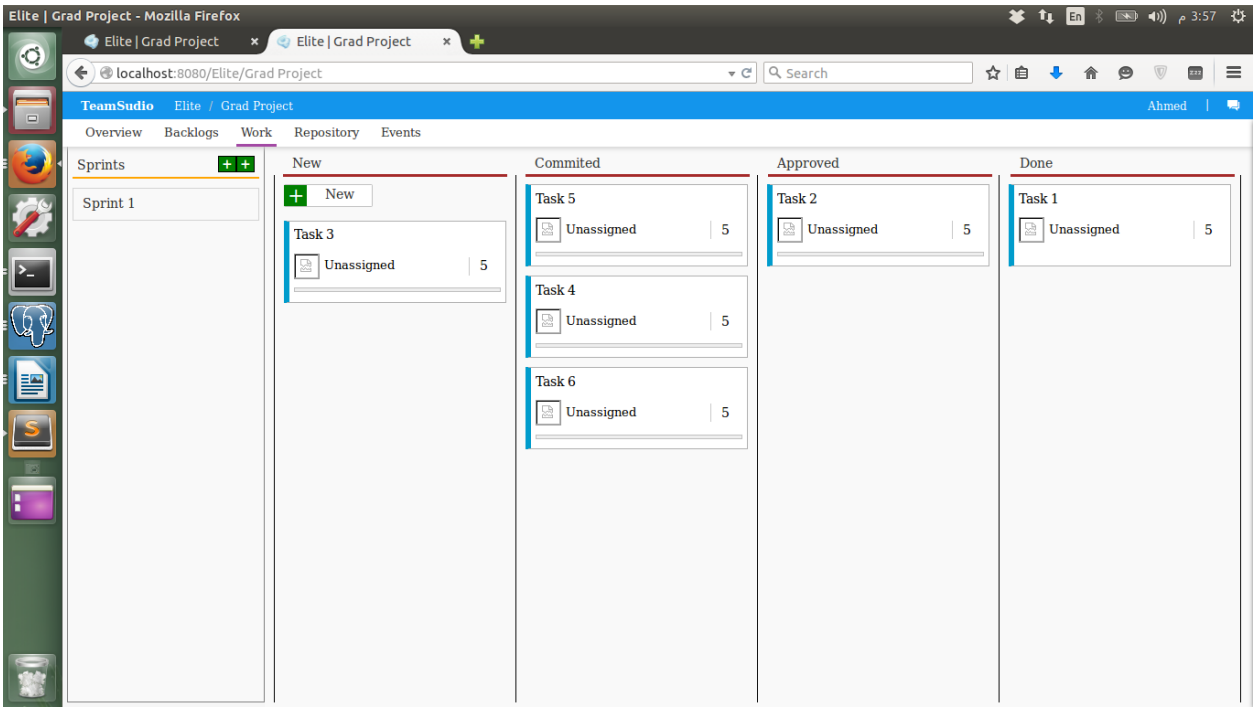


Figure 26 view tasks

The user can change task details by pressing the edit button, which is located on the task itself, a prompt will appear to fill in the new data of the edited task then the user can save the changes or exit without saving.

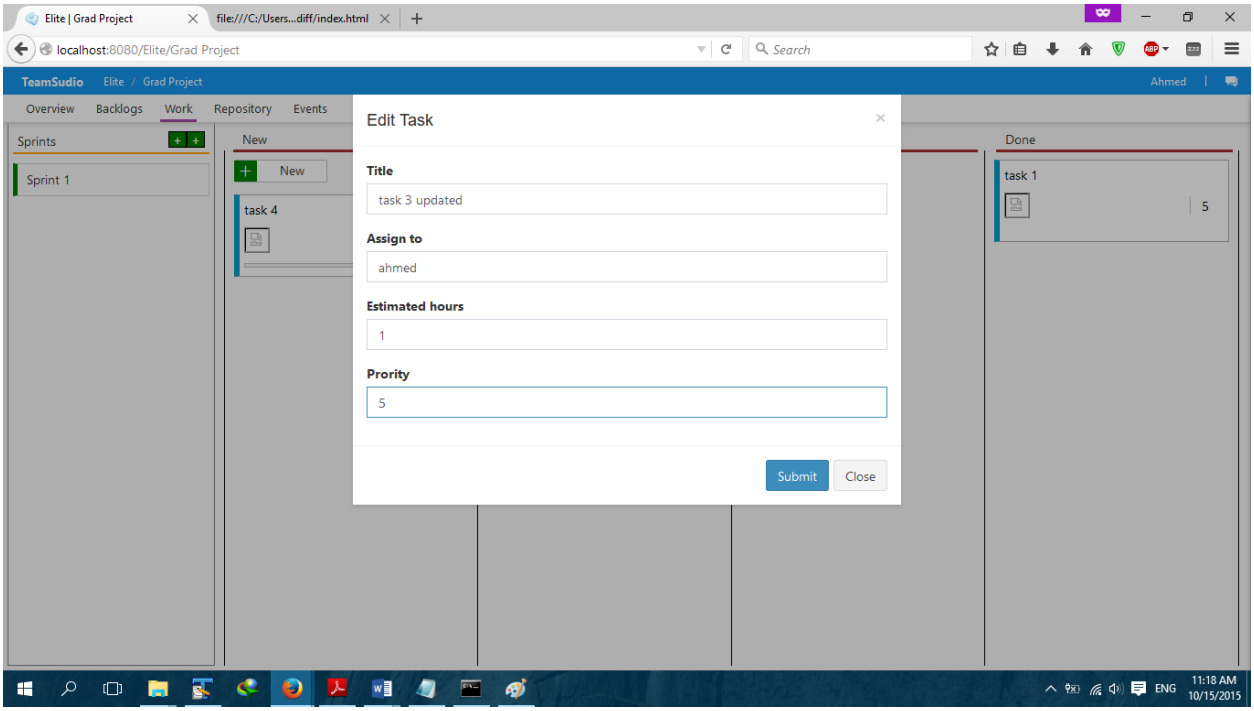


Figure 27 update task

The Repository tab is the place where you can view the source code from anywhere.

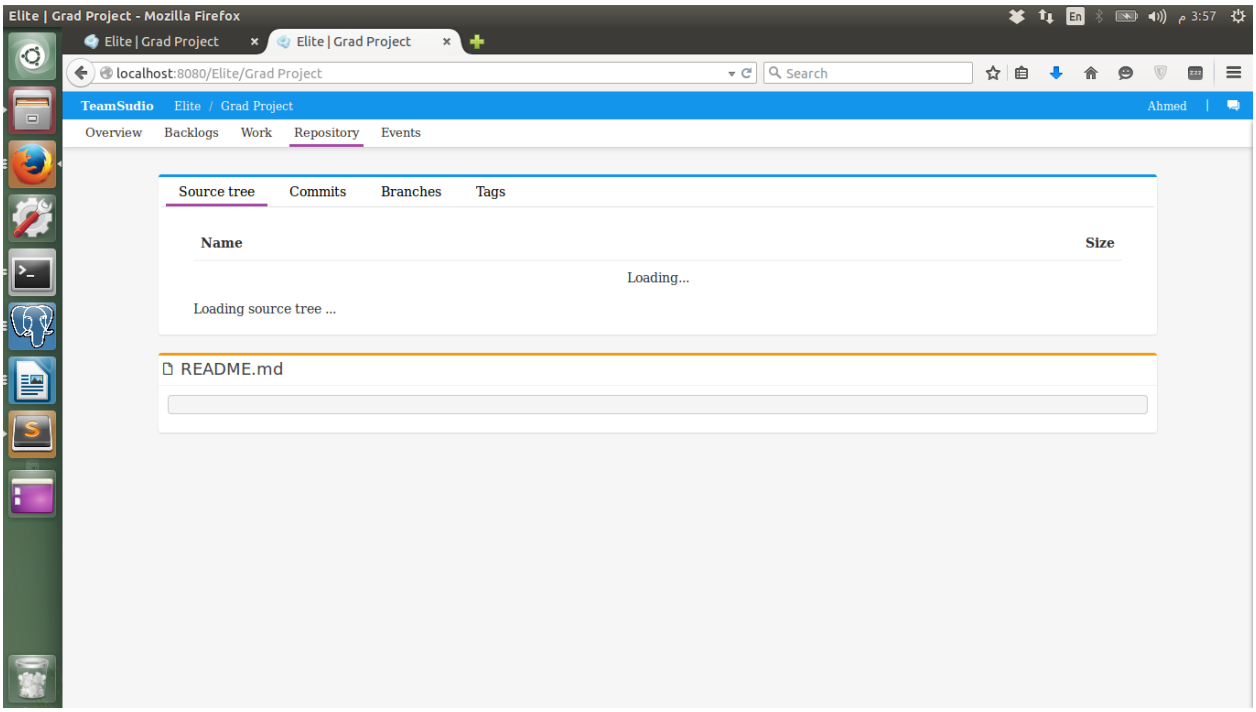


Figure 28 source tree is loading

This is a created project tree, where the user can view all the files and folders from this tab for a quick revision or inspection if needed.

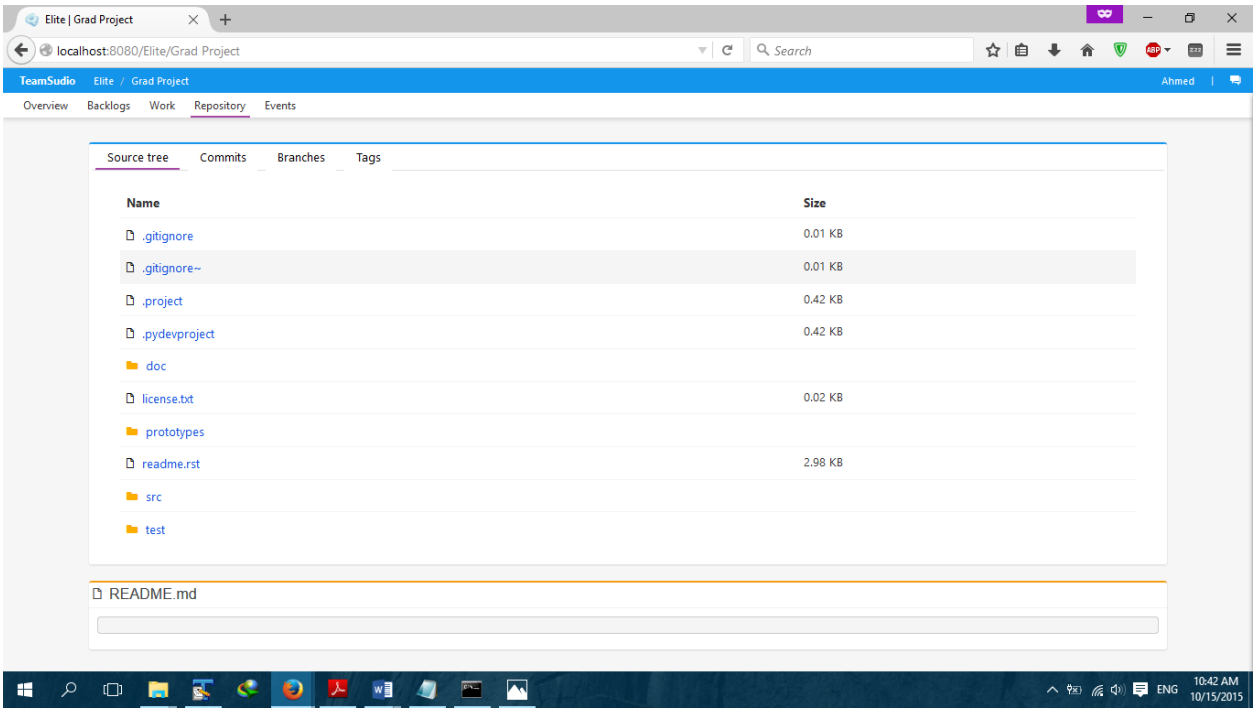


Figure 29 project source tree

This is the commit page, it shows who committed what and when with a certain ID for referencing or regression if needed and with the message left by the developers.

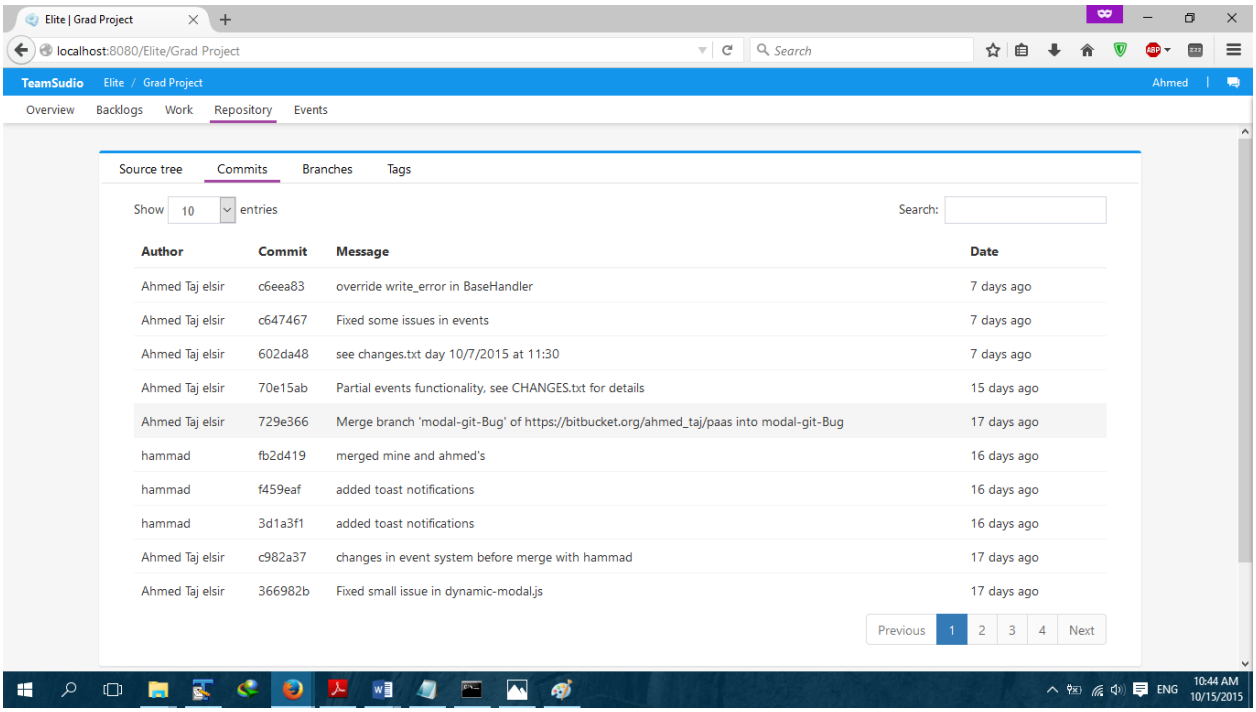


Figure 30 view project commits

This shows the difference between commits when a users clicks on a commit.

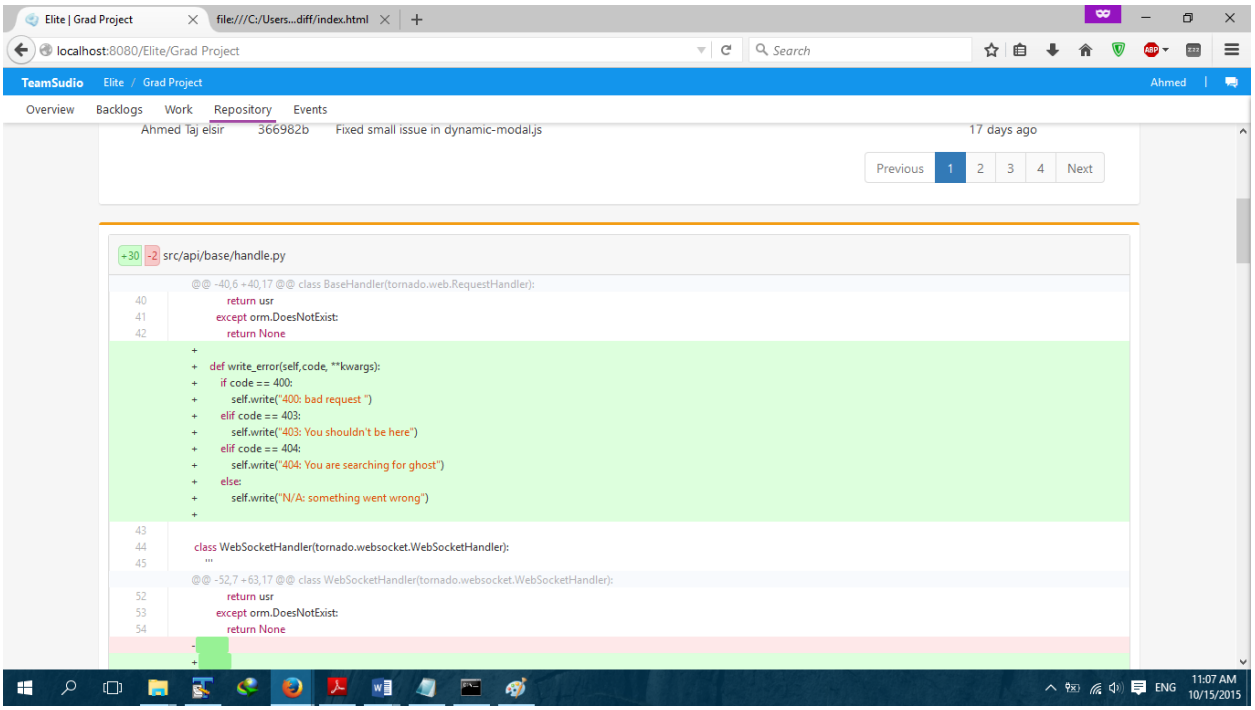


Figure 31 view code differences between commits

These are the version control branches you view them from the repo tab.

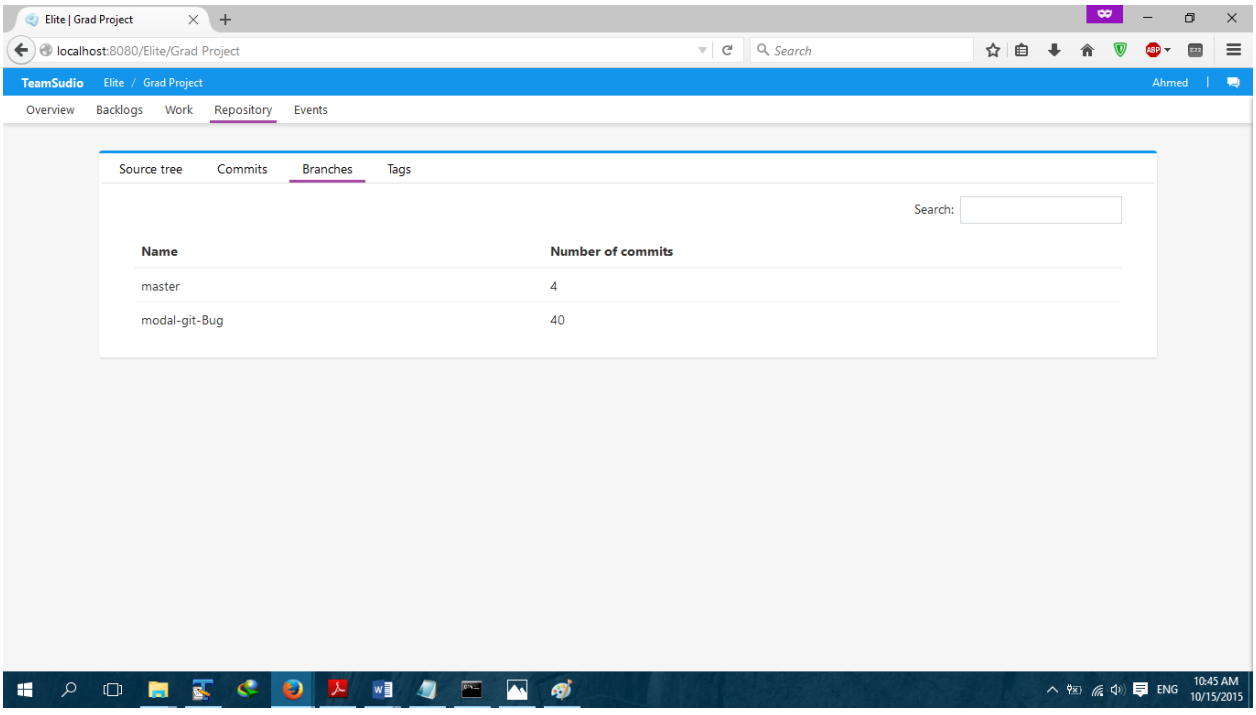


Figure 32 view branches

These are the created repository tags for advanced version control.

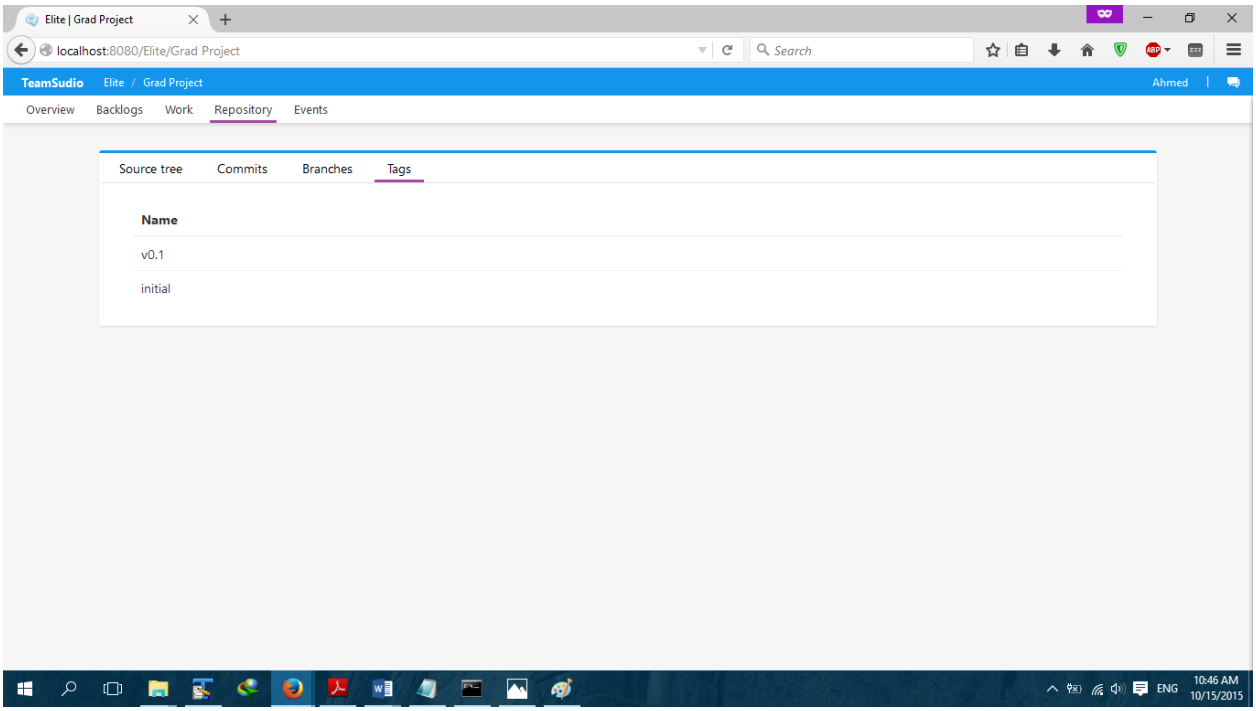


Figure 33 view tags

The user can edit and run code from our IDE by prompting us to run a virtual machine for him.

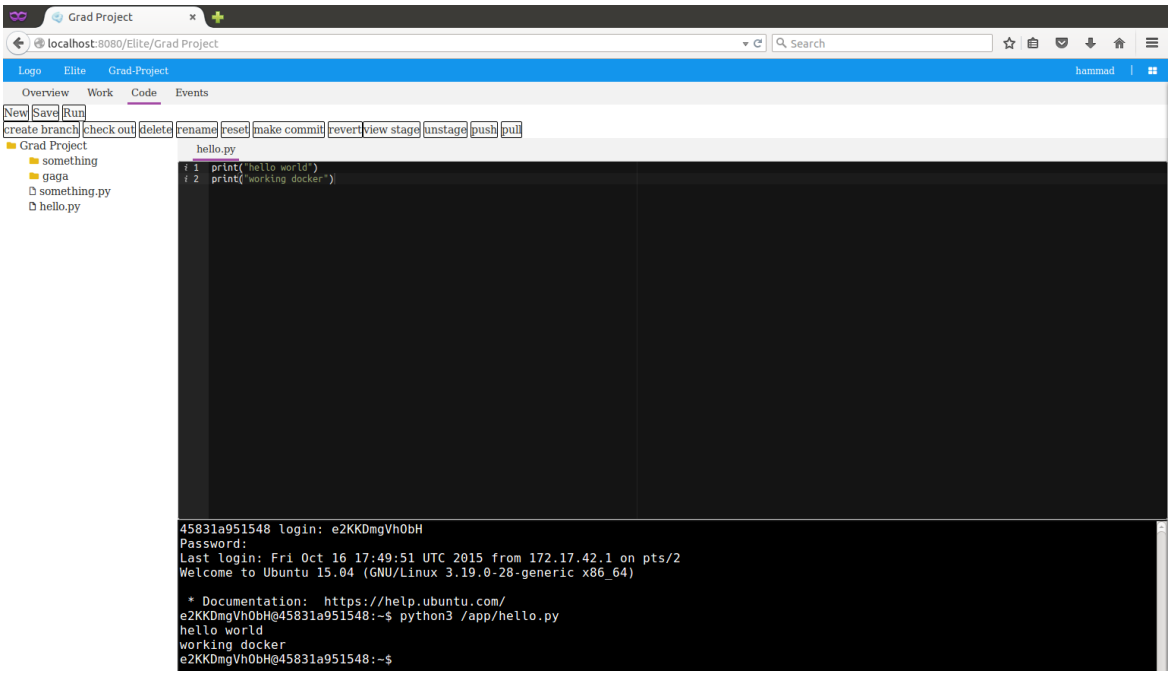


Figure 34 edit and run code

The events tab, here users can create meetings, todos and milestone reminders and events.

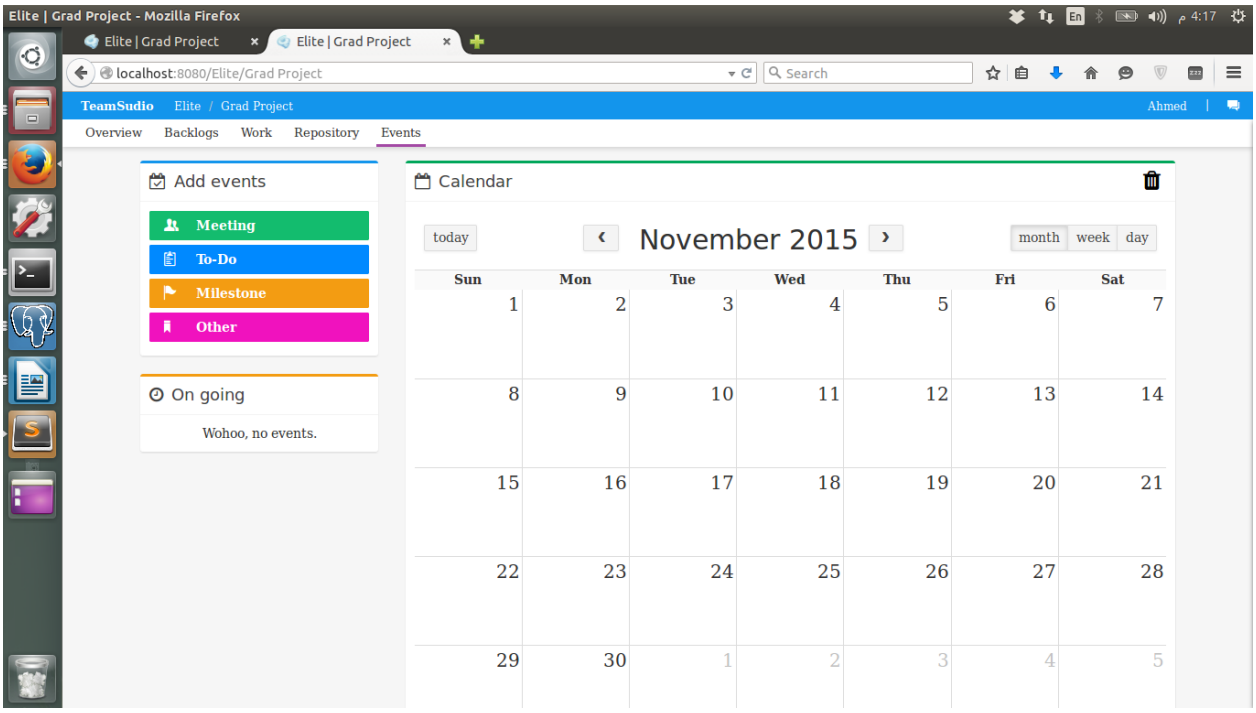


Figure 35 view project calendar

This is the create event modal, here a user can create an event by dragging the type of event into the date he wants the date created on and assign all the details that are needed to create an event.

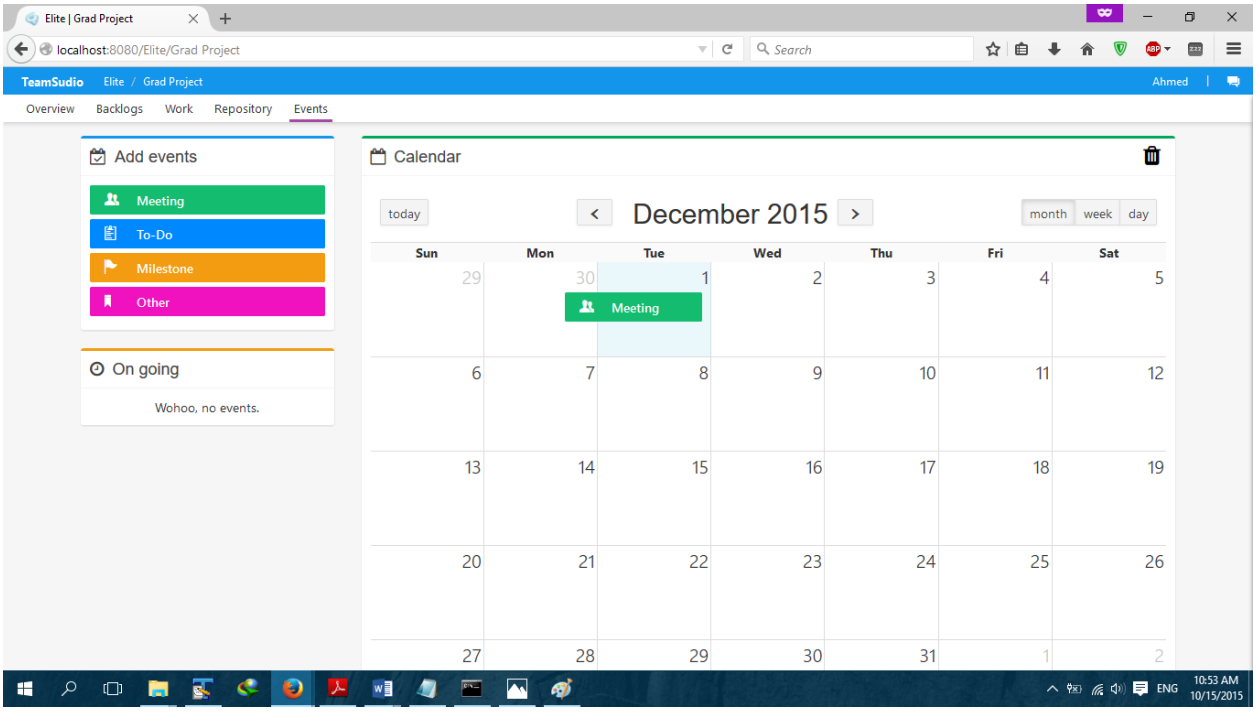


Figure 36 drag to add event

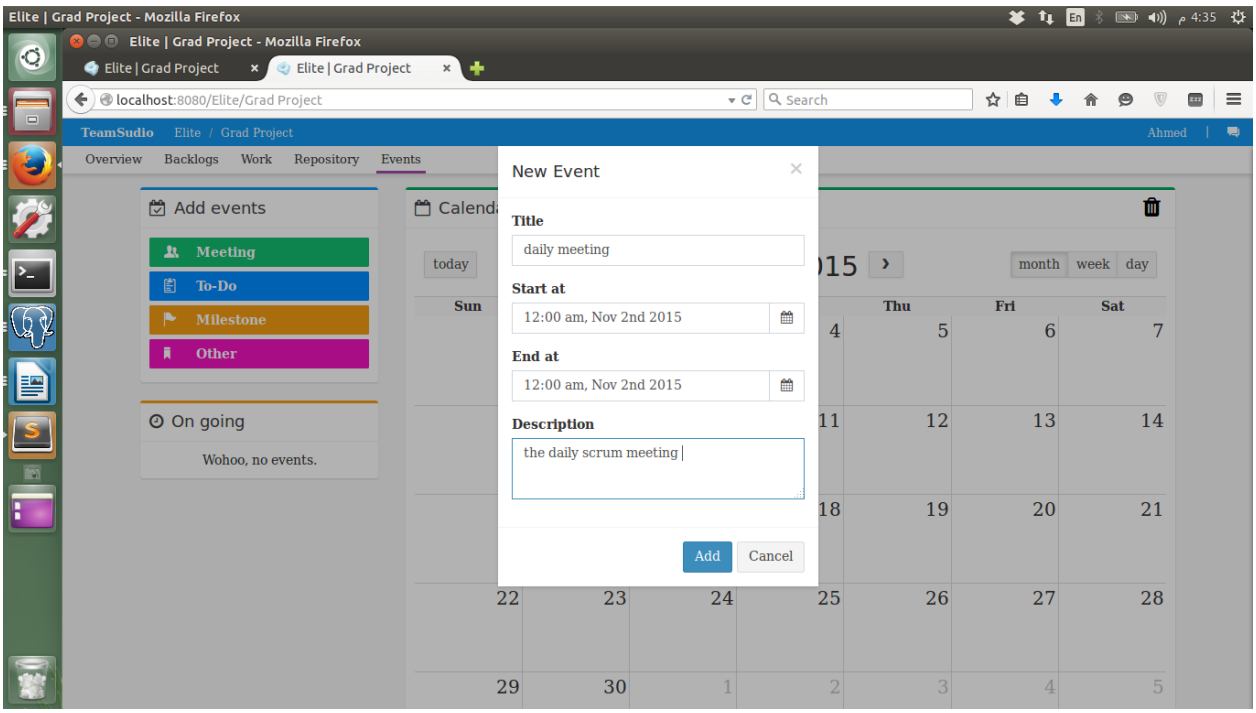


Figure 37 add event

After an event is a created the user will get a notification of the operation status.

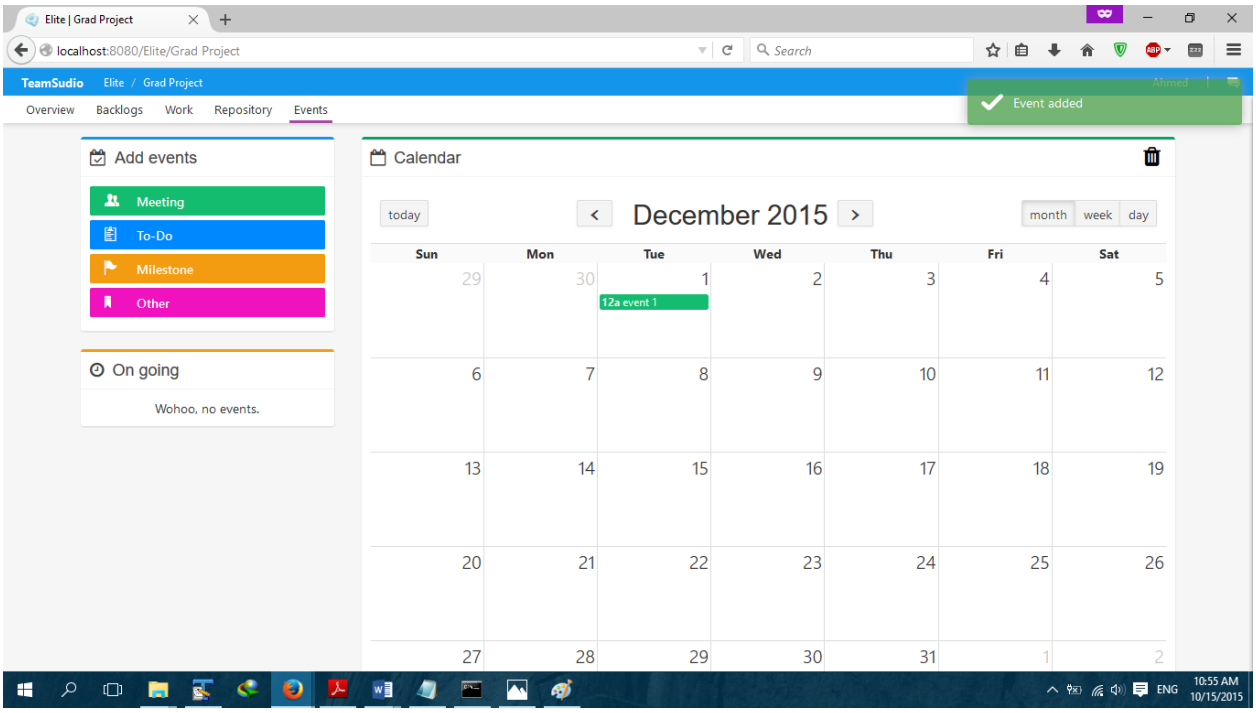


Figure 38 event added

The user can have quick access to the event details by just hovering over a certain event.

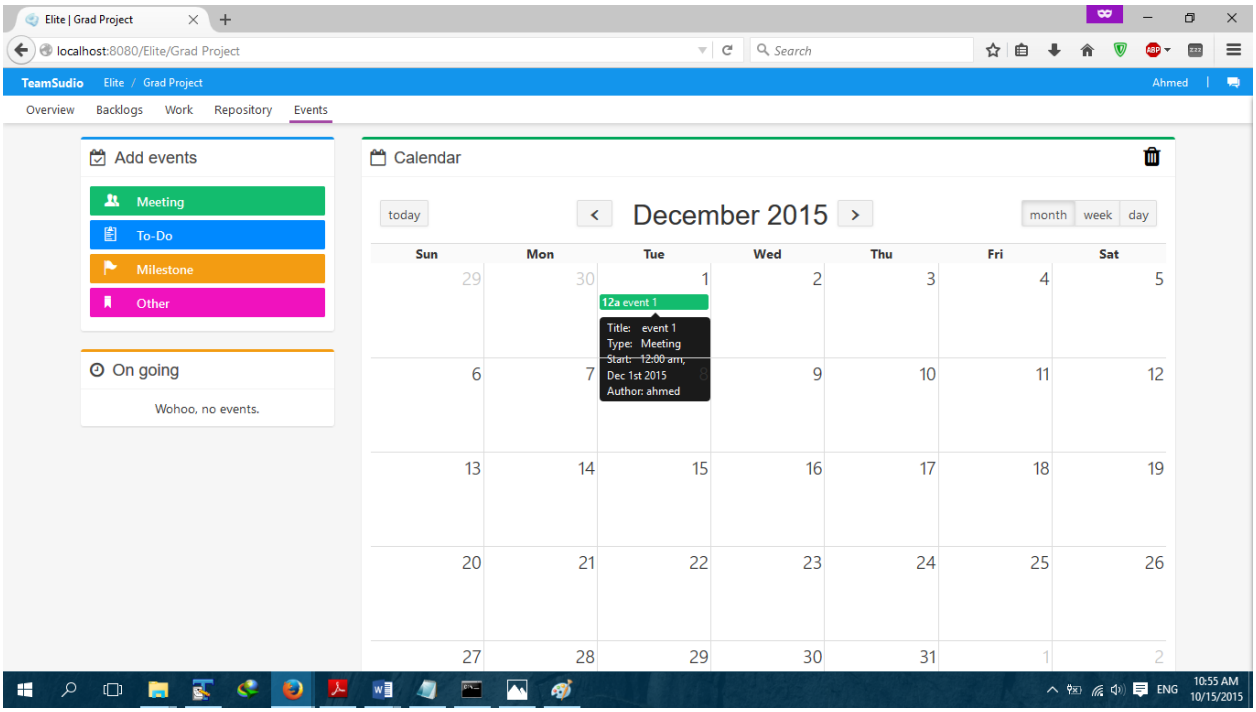


Figure 39 event details

If a user wishes to change event data he'll be prompted to enter the new event data in this screen.

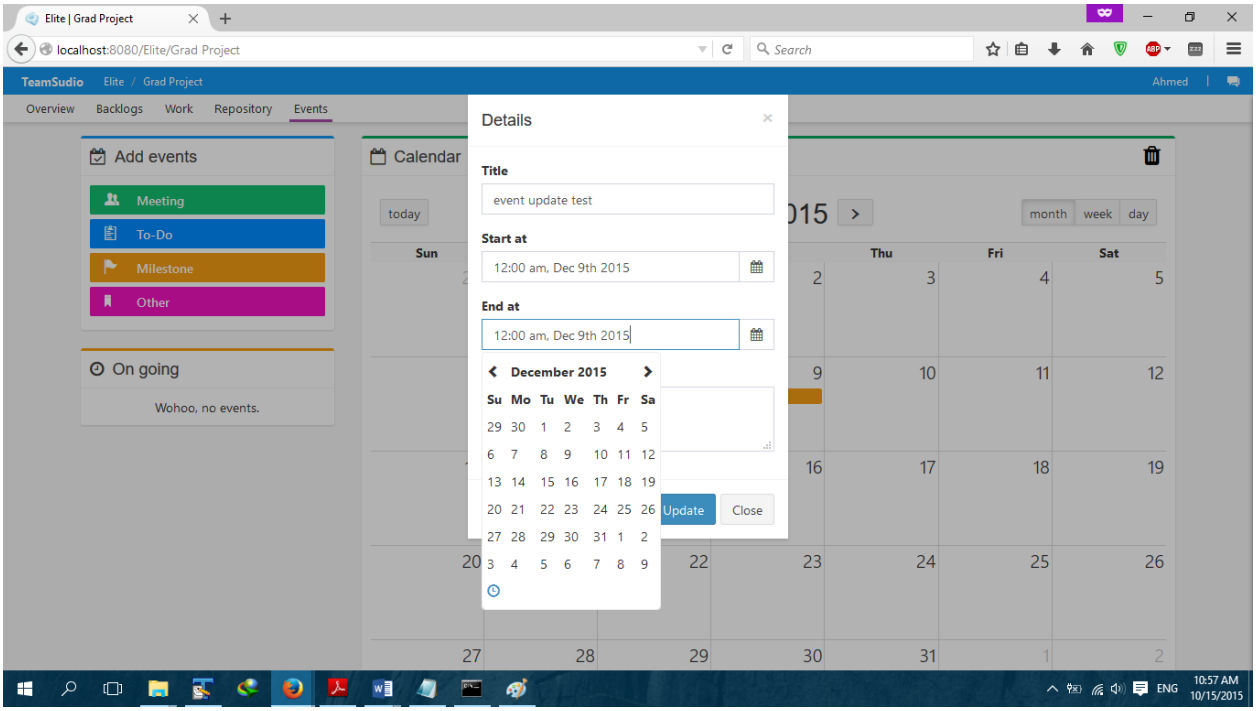


Figure 40 update event

When the user wants to delete an event all he needs to do is to drag it to the trashbin.

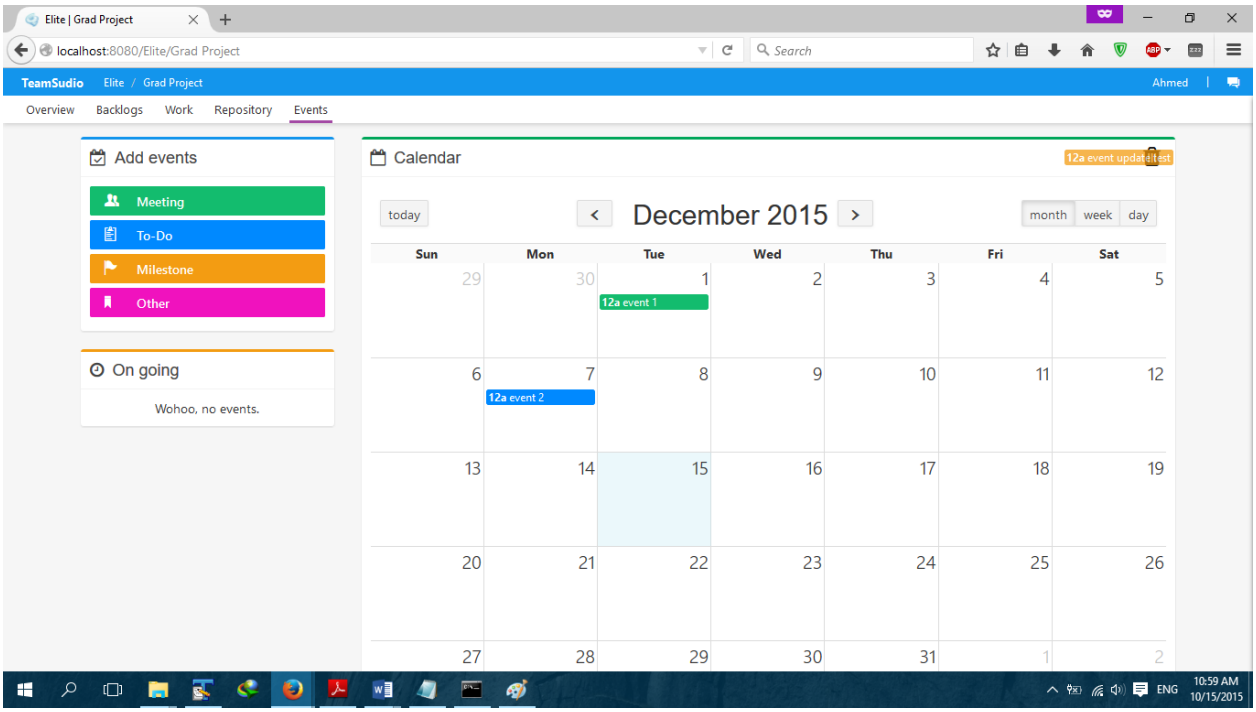


Figure 41 remove event

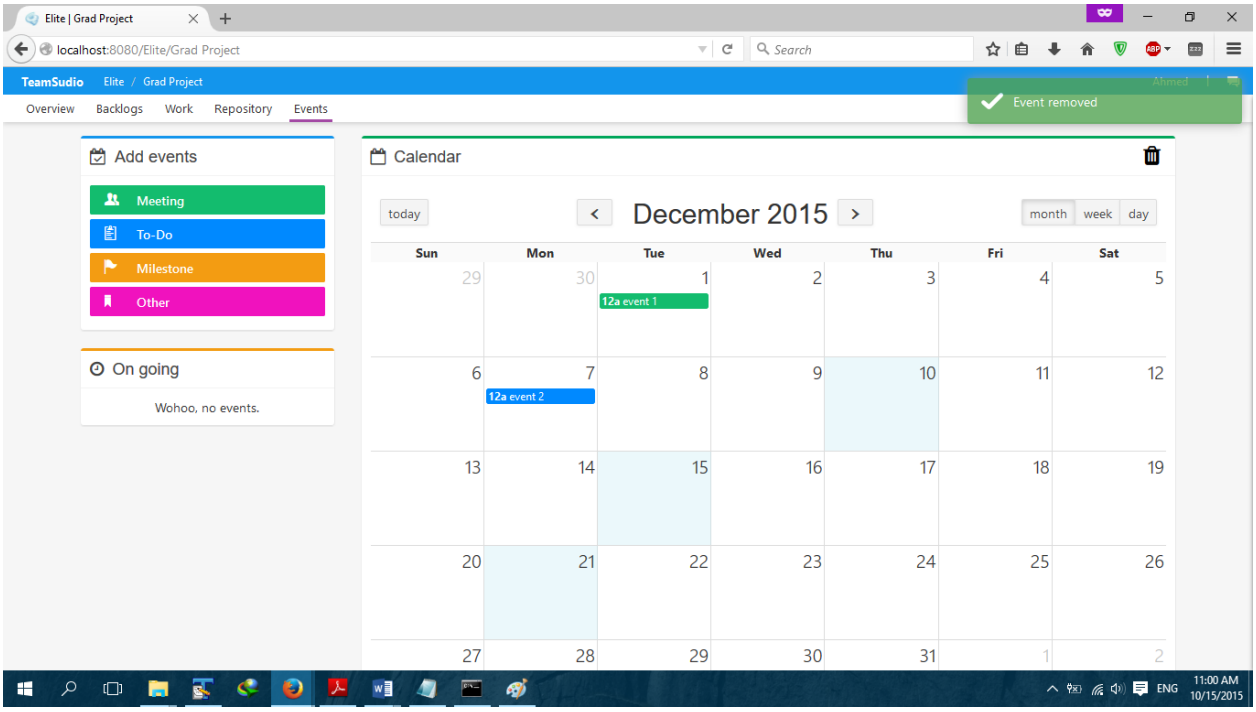


Figure 42 event removed

After a meeting is scheduel, users can enter the chat room and have a video conversation.

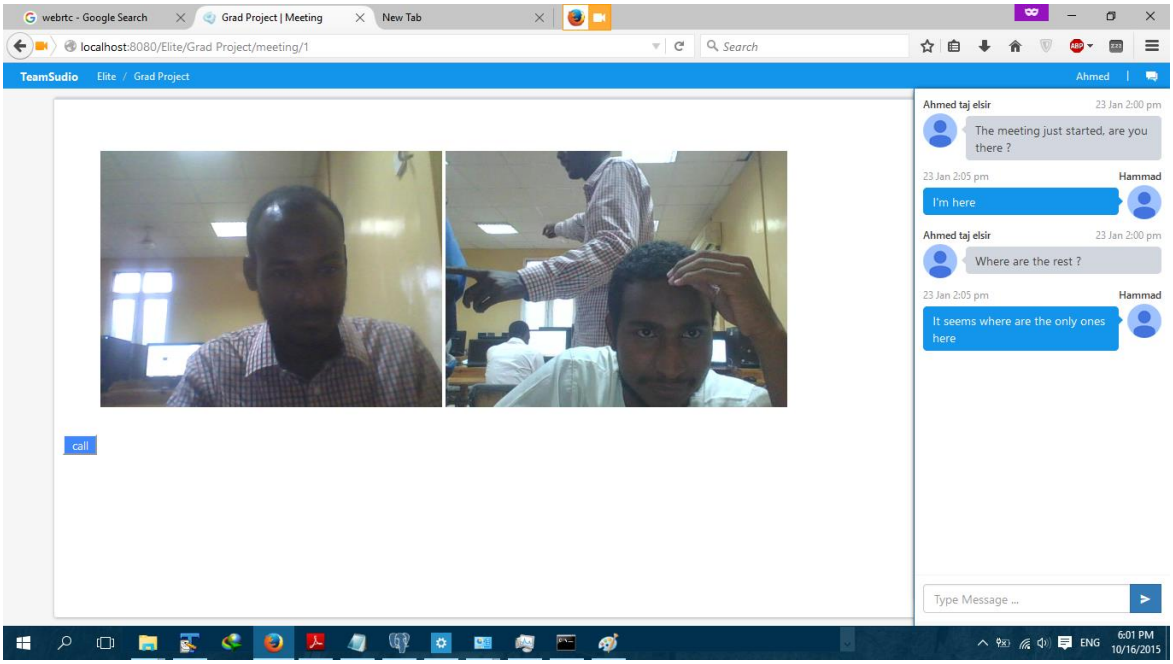


Figure 43 video meeting

5.4 TESTING:

Testing is a main part of Quality Aussrance of any software, the best method of testing is automated testing specially with an incremental model such that regresive tests are easier and no functionality is lost between increments.

There have been plans of a System wide automated test suite, such that all the functionality of the system will be easliy tested at every change, due to time constraints only a small section of the system had an actuall automated test.

We have created the four basic tests for the Task management to be automated.

The first test is used to add tasks and insure that they are inserted correctly.

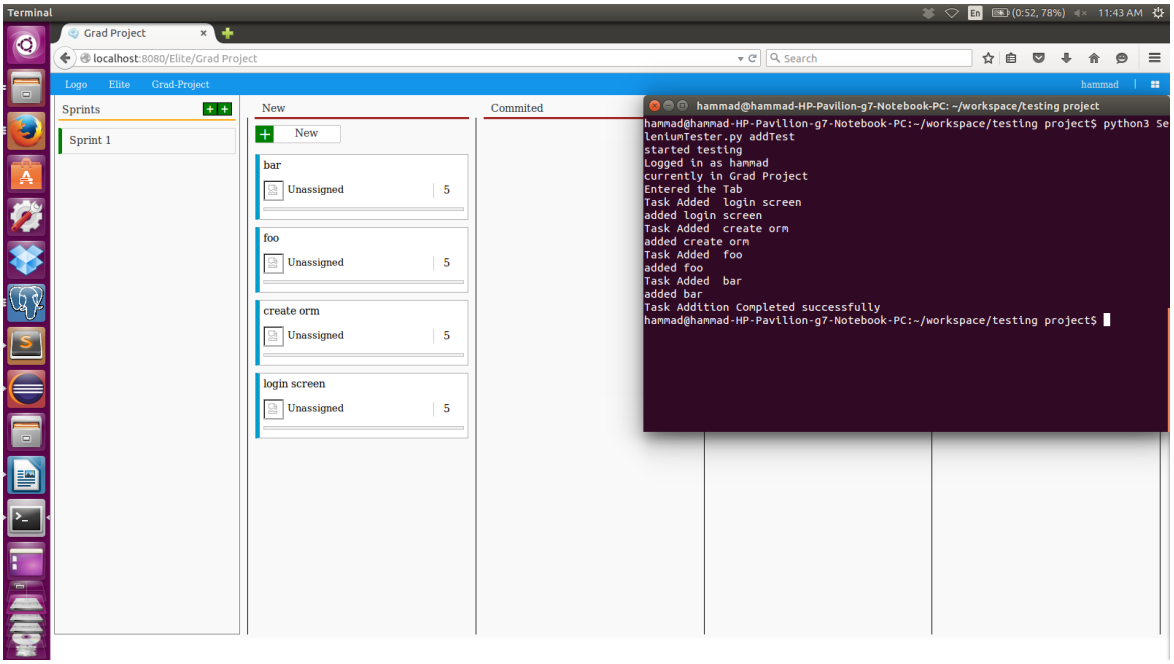


Figure 44 Task Addition Test

The second tests move tasks between columns

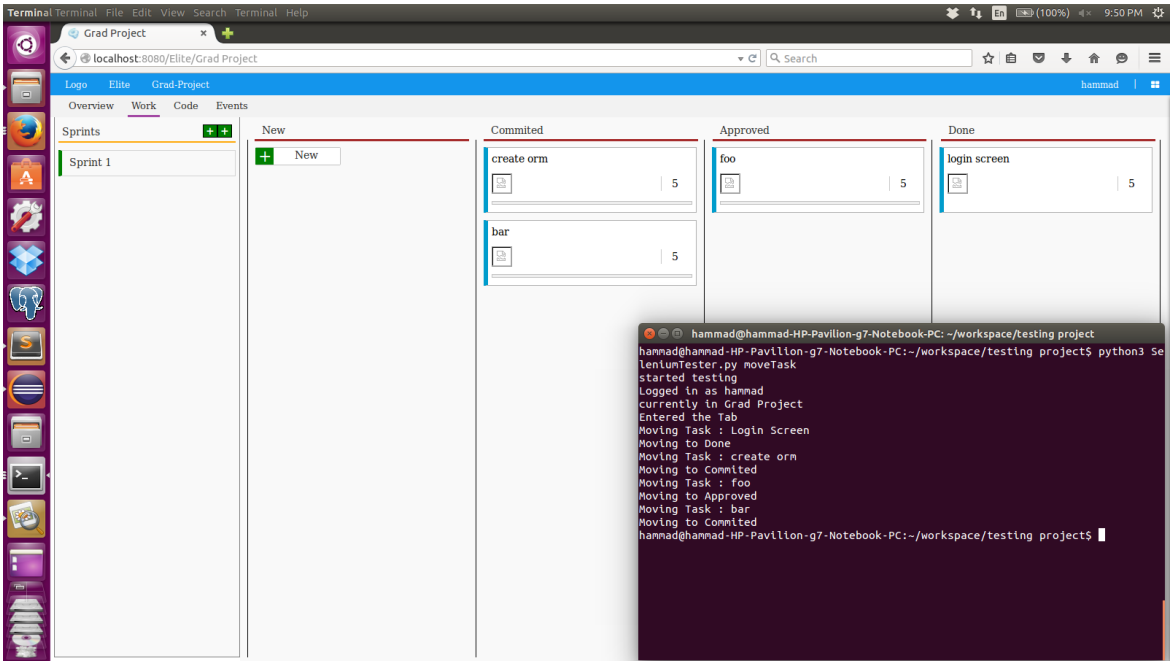


Figure 45 Task Drag Test

The third tests edits from of the tasks details

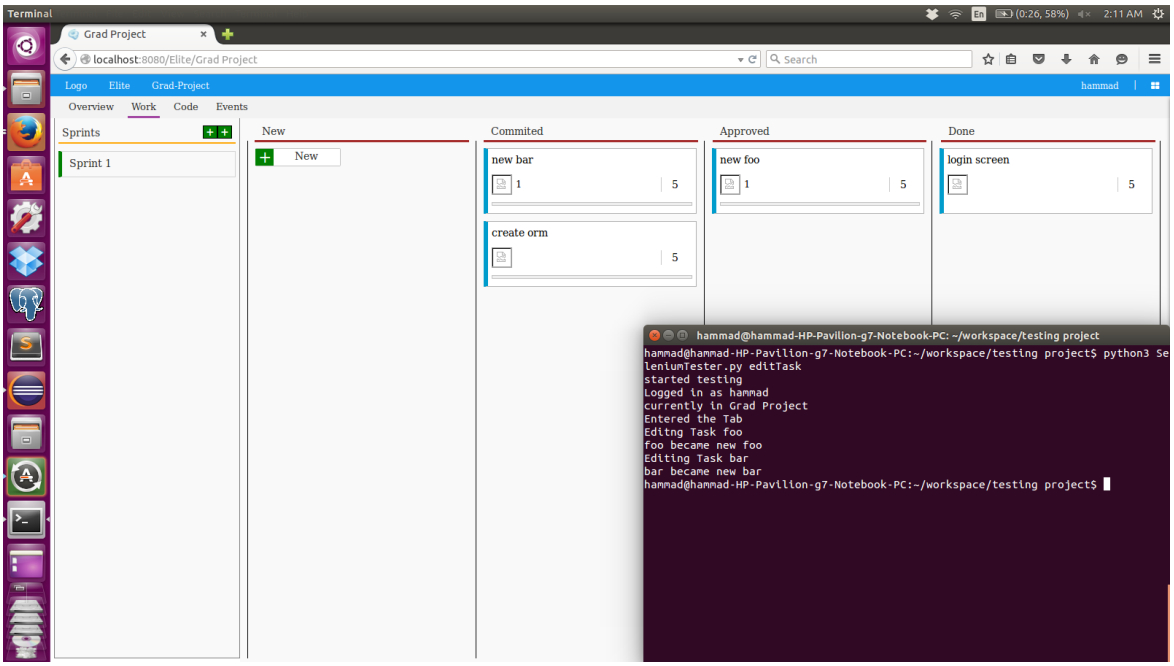


Figure 46 Task Edit Test

The Fourth tests is to delete tasks

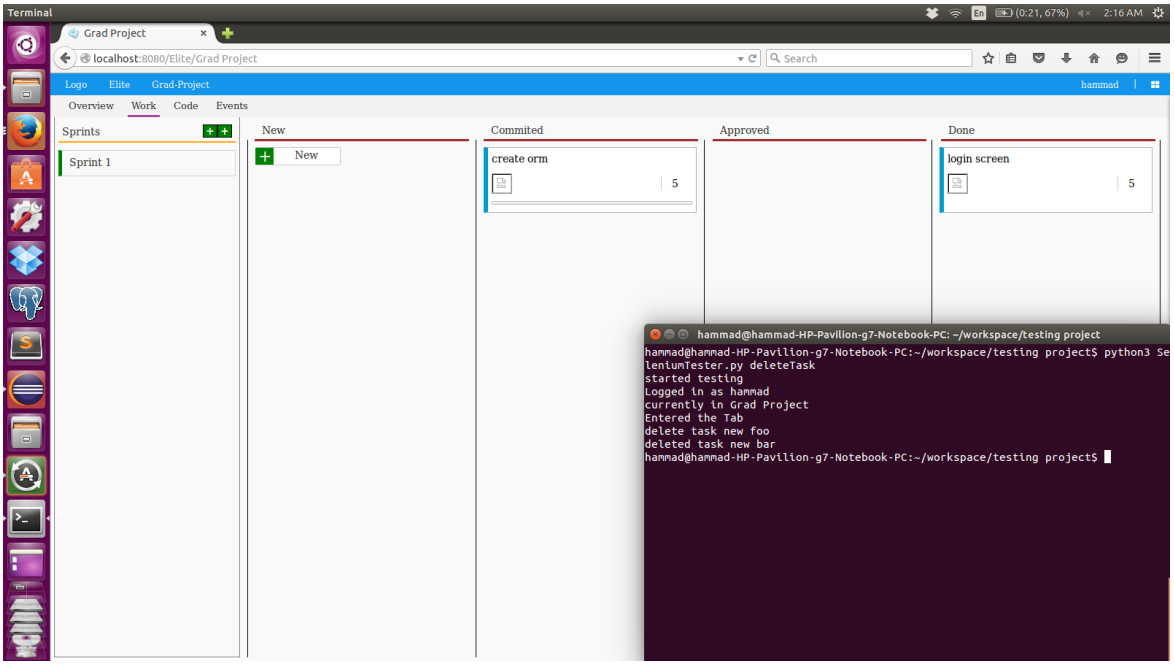


Figure 47 Task Deletion Test

CHAPTER SIX

RESULTS & RECOMMENDATION

6.1 INTRODUCTION:

This chapter discusses the results of the system, recommendations and obstacles that faced the project.

6.2 RESULTS:

- Ease of project management.
- Higher task management and better task assignment.
- Easier code access and sharing.
- Ability to develop software anywhere.
- Enhanced communication.
- Increased overall visibility.
- Improved resource allocation.
- Increased team autonomy.
- Control over product features.
- Decrease of development cost and required resources.
- Decrease in required effort for project setup.
- Increase in team efficiency.

6.3 OBSTACLES:

There were many obstacles in the creation of this project which are listed below:

- The large scope of the project.
- The amount of required background information and analysis.
- Major delay in providing required resources.
- Difficulty in configuring servers.

6.4 RECOMMENDATIONS:

All that we have accomplished is not but a small part of the whole picture, we hope that developers will continue where we stopped and perfect this project. The project needs improvement in many aspects, some of the most required features are:

- Wiki page for documentation
- Screen Sharing
- Full Automated tests
- UML design tools
- VC access level to who can push code
- Turning the project into an easy to deploy cloud arch
- The ability to view other members unfinished code

6.5 CONCLUSION

Software isn’t perfect, perfection isn’t something humans can accomplish but we as developers did our best to provide all the features we could that is required in a software of this scale, there has been a lot of effort to correctly complete this project.

The project aimed to ease and increase development of software, full implementation and perfection of this project would technically make that possible and our wishes is for this platform to prosper and develop.

REFERENCES

REFERENCES:

[1] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland and Dave Thomas, "Manifesto for Agile Software Development," 8 10 2015. [Online]. Available: http://www.agilemanifesto.org/. [Accessed 8 10 2015].

[2] A. k. Szo˝ke, "A Feature Partitioning Method for Distributed Agile," in *Agile Processes in Software Engineering and Extreme Programming*, Madrid, Spain, Springer-Verlag Berlin Heidelberg, 2011, pp. 27-42.

[3] S. Ambler, "Survey Says: Agile Works in Practice," *Dr. Dobb's Journal* , 2006.

[4] Tore Dybå and Torgeir Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and Software Technology*, vol. 50 , no. 9-10, pp. 833-859 , 2008.

[5] Kent Beck and Cynthia Andres, Extreme Programming Explained: Embrace Change, 2 ed., Addison-Wesley Professional, 2004.

[6] Ken Schwaber and Mike Beedle, Agile Software Development with Scrum, 1 ed., Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[7] "Front Page | DSDM CONSORTiUM," [Online]. Available: http://www.dsdm.org/. [Accessed 7 2015].

[8] Michele Marchesi and Katiuscia Mannaro, *Adopting Agile Methodologies in Distributed Software Development*, 2008.

[9] Thomas Stober and Uwe Hansmann, Agile Software Development, Springer-Verlag Berlin Heidelberg, 2010.

[10] K. S. Rubin, Essential Scrum: A Practical Guide to the Most Popular Agile Process.

[11] C. G. Cobb, The Project Manager's Guide to Mastering Agile: Principles and Practices for an Adaptive Approach.

[12] M. Cohn, succeeding with agile software development using scrum.

[13] P. L. Bannerman, E. Hossain and R. Jeffery, "Scrum Practice Mitigation of Global Software Development Coordination Challenges: A Distinctive

Advantage?," *HICSS '12 Proceedings of the 45th Hawaii International Conference on System Sciences*, pp. 5309-5318, 2012.

[14] Emam Hossain, Paul L. Bannerman and D. Ross Jeffery, "Scrum Practices in Global Software Development: A Research Framework," *12th International Conference, PROFES*, pp. 88-102, 2011.

[15] L. Williams, A Surrvey of Agile Development Methodologies.

[16] Miguel Jiménez, Mario Piattini and Aurora Vizcaíno, "Challenges and improvements in distributed software development: a systematic review," *Advances in Software Engineering*, 2009.

[17] Nilay Oza, Jürgen Münch, Juan Garbajosa, Agustin Yague and Eloy Gonzalez Ortega, "Identifying Potential Risks and Benefits of Using Cloud in Distributed Software Development," in *Product-Focused Software Process Improvement*, Springer-Verlag Berlin Heidelberg, 2013, pp. 229-239.

[18] *. H. H. O. P. J. Å. a. F. Eoin Ó Conchúir1, "Benefits of Global Software Development : Exploring the unexplored".

[19] U. Venkatesh, Distributed Agile: DH2A - The Proven Agile Software Development Approach and Toolkit for Geographically Dispersed Teams, echnics Publications, LLC, 2011.

[20] R. Jeffries, A. Anderson and C. Hendrickson, Extreme Programming Installed, Upper Saddle River, NJ: Addison Wesley, 2001.

[21] A. V. B. P. Jeff Sutherland, "Distributed Scrum: Agile Project Management with Outsourced Development Teams," p. 274a.

[22] wikipedia, "(HTML)-wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Bootstrap_%28front-end_framework%29. [Accessed 25 6 2015].

[23] D. Gandy, "fontawesome.io," [Online]. Available: http://fontawesome.io/.

[24] "Awesomplete: Ultra lightweight, highly customizable, simple autocomplete, by Lea Verou," [Online]. Available: https://leaverou.github.io/awesomplete/. [Accessed 6 2015].

[25] "pygit2’s documentation!," [Online]. Available: http://www.pygit2.org/. [Accessed 6 2015].

[26] "Nginx - Wikipeida," [Online]. Available: https://en.wikipedia.org/wiki/Nginx. [Accessed 6 2015].

[27] "Bootstrap (font-end framework) - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Bootstrap_%28front-end_framework%29. [Accessed 6 2015].

[28] "bootstrap-W3S," [Online]. Available: www.w3schools.com/bootstrap/bootstrap_get_started.asp.

[29] "Ace - The High Performance Code Editor for the Web," [Online]. Available: http://ace.c9.io. [Accessed 6 2015].

[30] "About - Git," [Online]. Available: http://git-scm.com/about. [Accessed 6 2015].

[31] "Chart.js | Open source HTML5 Charts for your website," [Online]. Available: http://www.chartjs.org/. [Accessed 6 2015].

[32] "nnnick/Chart.js · GitHub," [Online]. Available: https://github.com/nnnick/Chart.js. [Accessed 6 2015].

[33] "Tornado (web server) - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Tornado_%28web_server%29. [Accessed 6 2015].

[34] "redis 2.10.3 : Python Package Index," [Online]. Available: https://pypi.python.org/pypi/redis. [Accessed 6 2015].

[35] "Redis - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Redis. [Accessed 6 2015].

[36] "pycrypto 2.6.1 : Python Package Index," [Online]. Available: https://pypi.python.org/pypi/pycrypto. [Accessed 6 2015].

[37] S. -. W. B. Automatio. [Online]. Available: http://www.seleniumhq.org/. [Accessed 6 2015].

[38] "SeleniumHQ/selenium - GitHub," [Online]. Available: https://github.com/SeleniumHQ/selenium. [Accessed 6 2015].

[39] "libgit2," [Online]. Available: https://libgit2.github.com/. [Accessed 6 2015].

[40] "Redis," [Online]. Available: http://redis.io. [Accessed 7 2015].

[41] "(Nginx)wikipedia.org," [Online]. Available: https://en.wikipedia.org/wiki/Nginx.

[42] "What is Docker," [Online]. Available: https://www.docker.com/whatisdocker. [Accessed 6 2015].

[43] "Docker (software) - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Docker_(software). [Accessed 6 2015].

[44] "google code shellinabox," [Online]. Available: http://code.google.com/p/shellinabox/ . [Accessed 12 7 2015].

[45] "DataTables | Table plug-in for jQuery," [Online]. Available: https://www.datatables.net/. [Accessed 6 2015].

[46] "CodeSeven/toastr · GitHub," [Online]. Available: https://github.com/CodeSeven/toastr. [Accessed 6 2015].

[47] "Unified Modeling Language (UML)," [Online]. Available: http://www.uml.org/. [Accessed 6 2015].

[48] "web RTC," [Online]. Available: https://en.wikipedia.org/wiki/WebRTC. [Accessed 2 9 2015].

[49] "Ubuntu Wiki," Canonical, [Online]. Available: https://en.wikipedia.org/wiki/Ubuntu_(operating_system). [Accessed 10 2015].

[50] D. Wells, "Extreme Programming: A gentle introduction," 10 8 2013. [Online]. Available: http://www.extremeprogramming.org. [Accessed 12 4 2015].

[51] Giancarlo Succi and Michele Marchesi, "The costs and benefits of pair programming," in *Extreme programming examined* , Boston, MA, USA, Addison-Wesley Longman Publishing Co., Inc., 2001, pp. 223-243 .

[52] Ilenia Fronza, Alberto Sillitti, Giancarlo Succi and Jelena Vlasenko, "Analysing the Usage of Tools in Pair Programming Sessions," in *Agile Processes in Software Engineering and Extreme Programming*, Madrid, Spain, Springer-Verlag Berlin Heidelberg, 2011, pp. 1-11.

[53] E. Hossain, M. A. Babar and H.-y. Paik, "Using Scrum in Global Software Development: A Systematic Literature Review," *ICGSE '09 Proceedings of the Fourth IEEE International Conference on Global Software Engineering* , pp. 175-184, 2009.

[54] T. Dingsøyr, S. Nerur, V. Balijepally and N. B. Moe, "A decade of agile methodologies," *Journal of Systems and Software,* vol. 85, no. 6, pp. 1213-1221 , 2012 .

[55] P Deemer, NKV Hazrati and GBR Benefield, The Distributed Scrum Primer, 2013.

[56] Maria Paasivaara, Sandra Durasiewicz and Casper Lassenius, "Using Scrum in Distributed Agile Development: A Multiple Case Study," *IEEE International Conference on Global Software Engineering* , pp. 195-204, 2009 .

[57] Keith Braithwaite and Tim Joyce, "XP expanded: distributed extreme programming," *International conference on Extreme Programming and Agile Processes in Software Engineering*, pp. 180-188, 2005.

[58] Monica Yap, "Implementing Distributed Extreme Programming: A Case Study," 2010.

[59] Elizabeth Woodward, Steffan Surdek and Matthew Ganis, A Practical Guide to Distributed Scrum, IBM Press, 2010.

[60] "Git (software) - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Git_%28software%29. [Accessed 6 2015].

[61] "Bootstrap Get Started," [Online]. Available: www.w3schools.com/bootstrap/bootstrap_get_started.asp. [Accessed 6 2015].

[62] "Font Awesome - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Font_Awesome. [Accessed 6 2015].

[63] "Font Awesome The iconic font and CSS toolkit," [Online]. Available: http://fontawesome.io/. [Accessed 6 2015].

[64] "Ace (editor)," [Online]. Available: https://en.wikipedia.org/wiki/ACE_%28editor%29. [Accessed 6 2015].

[65] "Flask (web framework)," [Online]. Available: https://en.wikipedia.org/wiki/Flask_%28web_framework%29. [Accessed 6 2015].

[66] "Performance of Flask, Tornado, GEvent, and their combinations.md · GitHub," [Online]. Available: https://gist.github.com/andreif/6088558. [Accessed 6 2015].

[67] "Django (web framework)," [Online]. Available: https://en.wikipedia.org/wiki/Django_%28web_framework%29. [Accessed 6 2015].

[68] "Eclipse Community Survey," 2014. [Online]. Available: https://www.eclipse.org/org/community_survey/SurveySummary_2014-public.xls. [Accessed 6 2015].

[69] "Selenium (software) - Wikipeida," [Online]. Available: https://en.wikipedia.org/wiki/Selenium_(software). [Accessed 6 2015].

[70] "LeaVerou/awesomplete · GitHub," [Online]. Available: https://github.com/LeaVerou/awesomplete. [Accessed 6 2015].

[71] "Welcome to redis-py's documentation!," [Online]. Available: https://redis-py.readthedocs.org/en/latest/. [Accessed 6 2015].

[72] "peewee — peewee 2.6.3 documentation," [Online]. Available: https://peewee.readthedocs.org. [Accessed 6 2015].

[73] M. F. SANNER, "PYTHON: A PROGRAMMING LANGUAGE FOR SOFTWARE," The Scripps Research Institute.

[74] "jQuery," [Online]. Available: http://jquery.com/. [Accessed 6 2015].

[75] "JScripters.com: developing a site with Javascript," [Online]. Available: http://www.jscripters.com/. [Accessed 6 2015].

[76] "W3Schools Online Web Tutorials," [Online]. Available: http://www.w3schools.com . [Accessed 6 2015].

[77] "PostgreSQL: About," [Online]. Available: http://www.postgresql.org/about/. [Accessed 6 2015].

[78] "PostgreSQL - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/PostgreSQL. [Accessed 6 2015].

[79] "shellinabox - Web based AJAX terminal emulator - Google Project Hosting," [Online]. Available: http://code.google.com/p/shellinabox/. [Accessed 6 2015].

[80] "Unified Modeling Language - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Unified_Modeling_Language. [Accessed 6 2015].

[81] l. williams, a survey of agile development methodologies.

[82] "www.pygit2.org," [Online]. Available: http://www.pygit2.org/. [Accessed 23 8 2015].

[83] "Font Awesome eikipedia.org," [Online]. Available: https://en.wikipedia.org/wiki/Font_Awesome.

[84] "datatables," [Online]. Available: https://www.datatables.net/. [Accessed 3 10 2015].

APPENDICES

6.6 APPENDIX I:

Questions:

1. How does the team divide the tasks between them?

2. How the do they assign each the task?

3. Do you prefer to work remotely or not locally? Why?

4. Can the current communication systems i.e. text/video chatting replace the need for face-to-face meetings?

5. Do you have any problem facing you in the current way?

6. Do you have any improvement you are willing to see in near future?

7. How does the code get shared between the team members?

8. How do you integrate you codes?

9. How do you test the project and many times do you do that per project life cycle?

10. How do you trace and manage the bugs in the project?

11. What IDE do you prefer? Why?

12. Have you ever used a project management tool? if yes then:

    a. Do you think it did the required job?

    b. How do you evaluate it?

    c. What are its pitfalls?