1.2.6 مقدمة :

فى هذا الفصل يتم وصف كامل للمشروع وخطواته بالأشكال الرسُومية، وشرح كل عمليات النظام من البداية وحتى النهاية، ويتم توضيح العمليات بإستخدام لغة النمذجه الموحدة .

2.2.6 لغة النمذجة الموحدة:

تُعرَّف لغة النمذجة الموحدة (UML) بأنها تدوين بياني يسمح برسم مخططات تمثل مفاهيم البرمجيات ، إذ يمكن استخدامها لرسم مخططات تمثل فضاء المسألة أو تصميماً مقترحاً للبرمجية أو حتى تحقيقاً مكتملاً للبرمجية [10].

1.2.2.6 أنواع مخططات (UML):

تتضمن لغة (UML) ثلاثة أنواع رئيسية من المخططات:

1. Static diagrams :

تصف المخططات السكونية للبنية المنطقية الثابتة للعناصر البرمجية بتصوير الصفوف والأغراض وبنى المعطيات والعلاقات الموجودة بينها [16].

2. Dynamic diagrams:

تبين المخططات الديناميكية كيف تتغير الكيانات البرمجية خلال التفيذ بتصوير تدفق التنفيذ أو الطريقة التي تغير بها الكيانات حالاتها [16].

3. Physical diagrams :

المخططات الفيزيائية تظهر البنية الفيزيائية الثابتة للكيانات البرمجية

بتصوير الكيانات الفيزيائية كالملفات المصدرية والمكتبات والملفات الثنائية وملفات المعطيات وغيرها ، والعلاقات الموجودة [10].

2.2.2.6 مميزات ال (UML):

1. تقدم اللغة وسيلة رمزية مبسطة للتعبير عن مختلف نماذج العمل البرمجي يَسهُل بواسطتها على المحللين والمصممين والمبرمجين العمل و التخاطب فيما بينهم وتمرير المعلومات في صيغة نمطية موحدة موجزة تغني عن الوصف العام باللغة العادية.
2. لغة النمذجة الموحدة (UML) ليست منهجية لبناء أو تصميم البرمجيات وتطويرها.
3. لاترتبط لغة البرمجة الموحدة(UML) بمنهجية أو طرق إنتاج البرمجيات.
4. لغة البرمجة الموحدة (UML) هي لغة نمذجة رسومية تقدم صيغة لوصف العناصر الرئيسية للنظم البرمجية.
5. بالرغم من بساطة وسهولة لغة النمذجة الموحدة لتوصيف وتمثيل النظم البرمجية إلا أنها قادرة على توصيف أعقد نظم المعلومات.
6. في لغة البرمجة الموحدة (UML) العديد من المخططات المختلفة (نماذج) ، ويرجع سبب التنوع إلى أن النظام الواحد يمكن النظر إليه من زوايا مختلفة بوجهات نظر مختلفة حسب المشاركين فيها.
7. تعتبر لغة النمذجة الموحدة لغة قياسية لتوصيف (Specifying) وتصوير (Visualizing) وتركيب (Constructing) وتوثيق (Documenting) نظم البرمجيات (Software System) بالإضافة إلى نمذجة الأعمال (Business Modeling) وغيرها من النظم غير البرمجية (Non-Software).
8. تقدم أفضل خبرات الممارسات الهندسية التي ثبت نجاحها في نمذجة النظم الكبيرة والمعقدة.
9. تزود المستخدمين بلغة جاهزة الإستخدام للنمذجة المرئية لتطوير وتبادل نماذج ذات معنى ولا تعتمد على لغة برمجة بعينها.
10. تعطي اللغة صورة كاملة عن البرنامج المراد تصميمه مما يسهل عملية تصور البرنامج كاملاً ويسهل صيانته.
11. يمكن توزيع المخططات على المبرمجين لإنجاز برنامج الأمر الذي يسرع عملية إنشاء البرنامج ، كما أنه في حالة الرجوع لهذه المخططات بعد فترة يسهل فهم البرنامج [10].

3.2.6 Enterprise Architecture (UML Model : Driven)

هي واحدة من أدوات هندسة البرمجيات بمساعدة الحاسوب (CASE) تستخدم في عملية تحليل و تصميم الأنظمة البرمجية وتعتمد في التصميم على النمذجة الموحدة .

هي إختصار للغة النمذجة الموحدة (The Unified Modeling Language)، وهي اللغة المستخدمة في الصناعة وتعتبر معياراً لدى مطوري ومصممي النظم وتعادل الرسم المعماري لدى مهندسي المباني، أي أنها طريقة موحدة للتعبير عن النظم، وتم اللجوء إلى هذا الأسلوب لأن التعبير عن النظم بإستخدام اللغات الطبيعية – (natural languages)- غير دقيق.

وتحتوي لغة الـ (UML) غالباً على رسوم غنية ومجموعة شاملة من المخططات والعناصر وجميعها واضحة وغير مُضللة ويتم فهمها بسهولة، ولهذه اللغة تاريخ طويل إلى أن أصبحت إحدي أقوى اللغات التي تساعد في تطوير النظم ولايمكن تخيل أي تصميم لنظام مسقبلي دون إستخدامها أو إستخدام لغة شبيهة بها، ولغة الـ (UML) تدعم 12 مخططاً بيانياً وهي:

- Package diagrams
- Class or Structural diagrams
- Object diagrams
- Composite Structure
- Component diagrams
- Deployment diagrams
- Use Case Diagrams
- Activity diagrams
- State Machine diagrams
- Communication diagrams
- Sequence diagrams
- Timing diagrams
- Interaction Overview diagrams
- Profile diagrams

لغة النمذجة الموحدة (UML) ظهرت في عام 1990م على أنها محاولة لإختيارأفضل العناصر من العديد من لغات النمذجة المقترحة في ذلك الوقت، والجمع بينهما وتكوين لغة واحدة متماسكة، فقد أصبحت منذ ذلك الوقت معياراً للصناعة لوضع النماذج والبرمجيات وتصميمها، وثم إستخدامها في مجالي العلم والتجارة [18].
أداة Enterprise Architect تدعم أحدث نسخة من لغة (UML 2.4.1) القياسية، على النحو الذي حددته مجموعة (Object Management Group (OMG)). وهذا يتيح لمستخدميه الوصول إلى جميع أنواع الرسم البياني المذكورة أعلاه، إضافة لإمكانية إستخراج مجموعة مختلفة من التقارير ليكون بيئة كاملة لإنتاج البرامجيات، وذلك لمساعدة المستخدمين على التواصل وتبادل النماذج الخاصة بهم على نحو أكثر فعالية.
Enterprise Archit هي أداة لنموذج تطويرالبرمجيات من إنتاج شركة (Sparx Systems) تحتوي علي واجهات رسومية سهلة الإستخدام إلى جانب دعمها لجميع الإمكانات التي توفرها لغة الـ (UML) وهي من الأدوات المستخدمة في الصناعة بقوة وتعتمدها العديد من شركات إنتاج البرمجيات المتطورة [11].

1.3.2.6 مميزات ال Enterprise Architecture :

1. يساعد في بناء وتوثيق قوي لصيانة النظم والعمليات .
2. يستخدم في تطوير مجموعة كبيرة من التطبيقات و الأنظمة .
3. يغطي مجالات متعددة مثل (الطيران ،الخدمات المصرفية ،الهندسة ،المالية ،الطيب ، العسكرية ، النقل ، تجارة التجزئية ، الهندسة الكهربئية ، الخدمات الاخرى).
4. إدارة جمع وتحليل المتطلبات .
5. تصميم الأنظمة بإستخدام لغة النمذجة الموحدة [11].

4.2.6 مخطط العمليات :Use case Diagram

هو عبارة عن مخطط يوضح كيفية تعامل النظام مع العالم الخارجي، كما أنه يصف سلوك النظام من وجهة نظر المُستخدم ويقصد به إمكانية أو وظيفة يقدمها النظام فاعل (Actor) معين مثلاً الدخول إلى النظام بإستخدام كلمة عبور أو إضافة سجل جديد إلى النظام [11].

فى هذا النظام يتم التعامل مع نوعين من أنواع المُستخدمين :

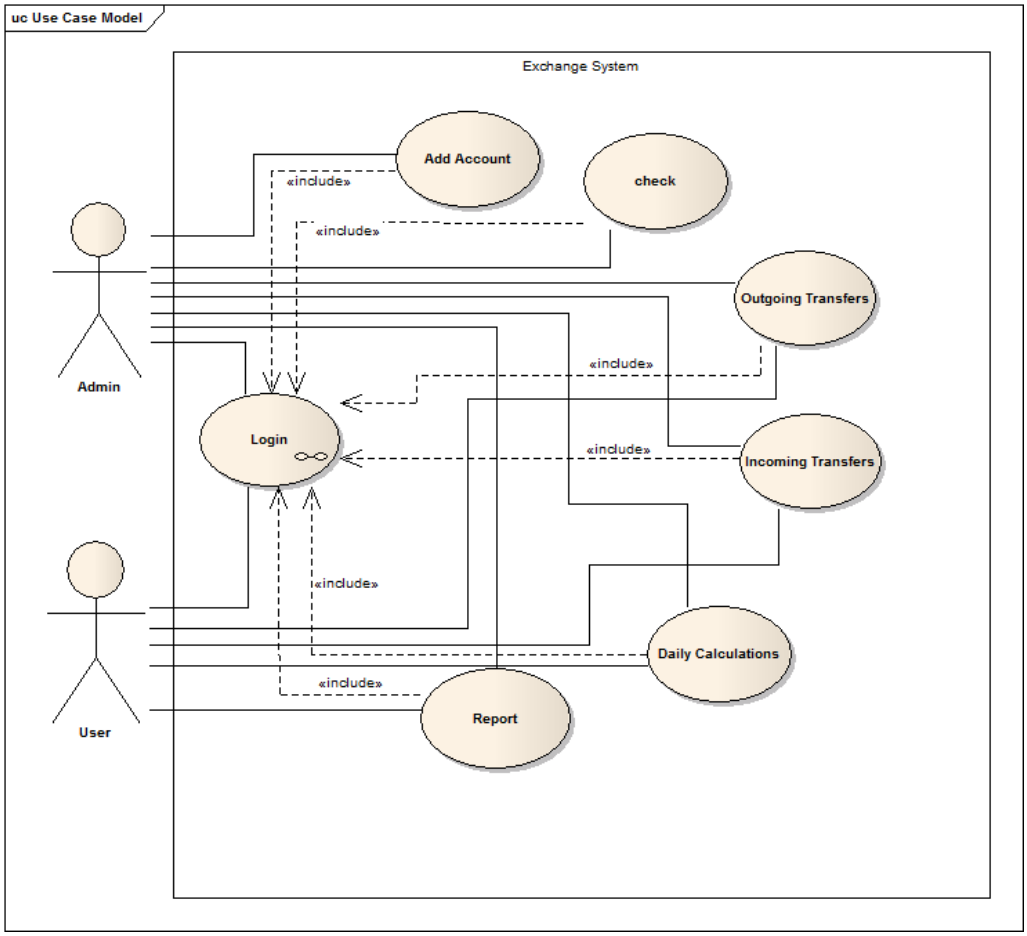
- مُدير نظام البنك و هيئة التوثيق :

هو الشخص الذ يقوم بإدارة الموقع، كما أنه المسؤول عن إدارة الشهادات الرقمية، ويقوم بالعمليات الآتية وذلك بعد تسجيل الدخول للنظام :-

- إضافة عميل.
- إجراء معاملات بنكية (حساب يومي، تحاويل واردة، تحاويل صادرة).
- التأكد من المعاملات التي تمت.
- إصدار التقارير عن المعاملات التي تمت.
- العميل :

يقوم باختيار العملية المراد إجراءها بعد تسجيل الدخول للنظام والتأكد من صحة اسم المسخدم وكلمة المرور.يقوم بالعمليات الآتية:

- إجراء معاملات بنكية (حساب يومي، تحاويل واردة، تحاويل صادرة).
- إصدار التقارير عن المعاملات التي تمت.

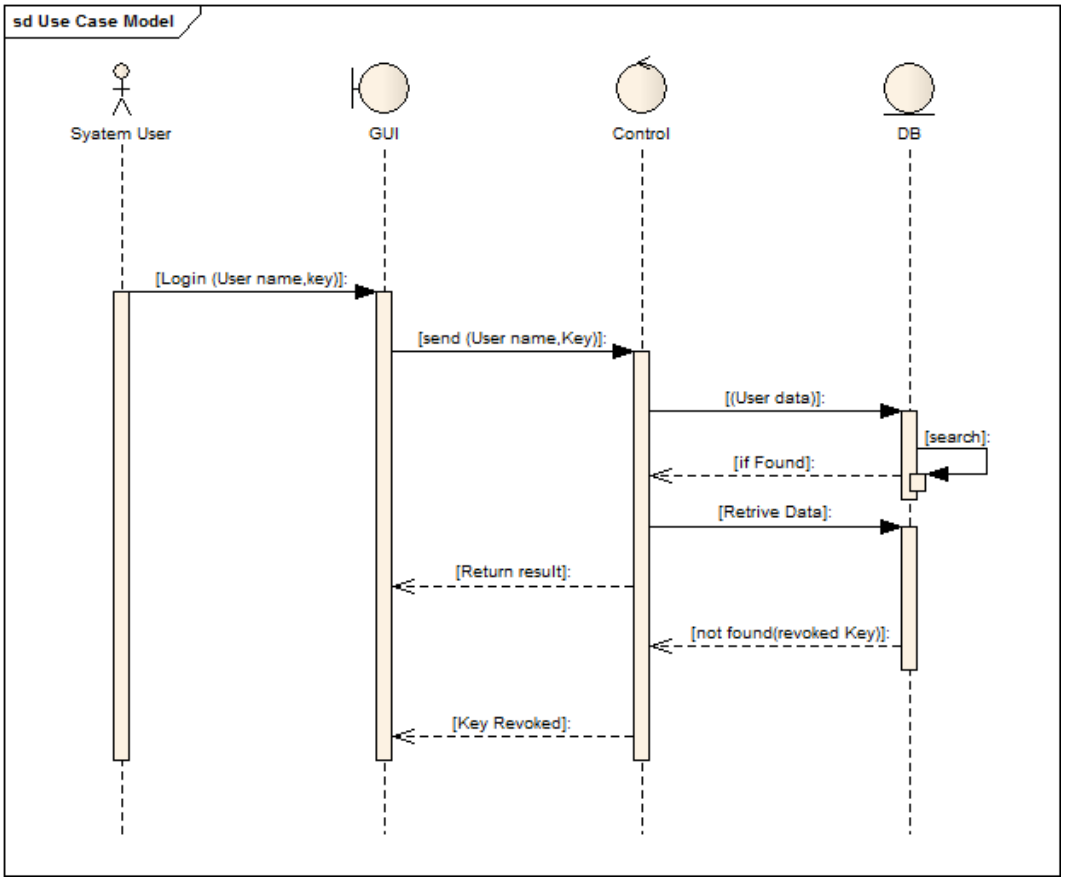


الشكل (1.6) : نظام الصرافة

5.2.6 مخطط التتابع : Sequence Diagram

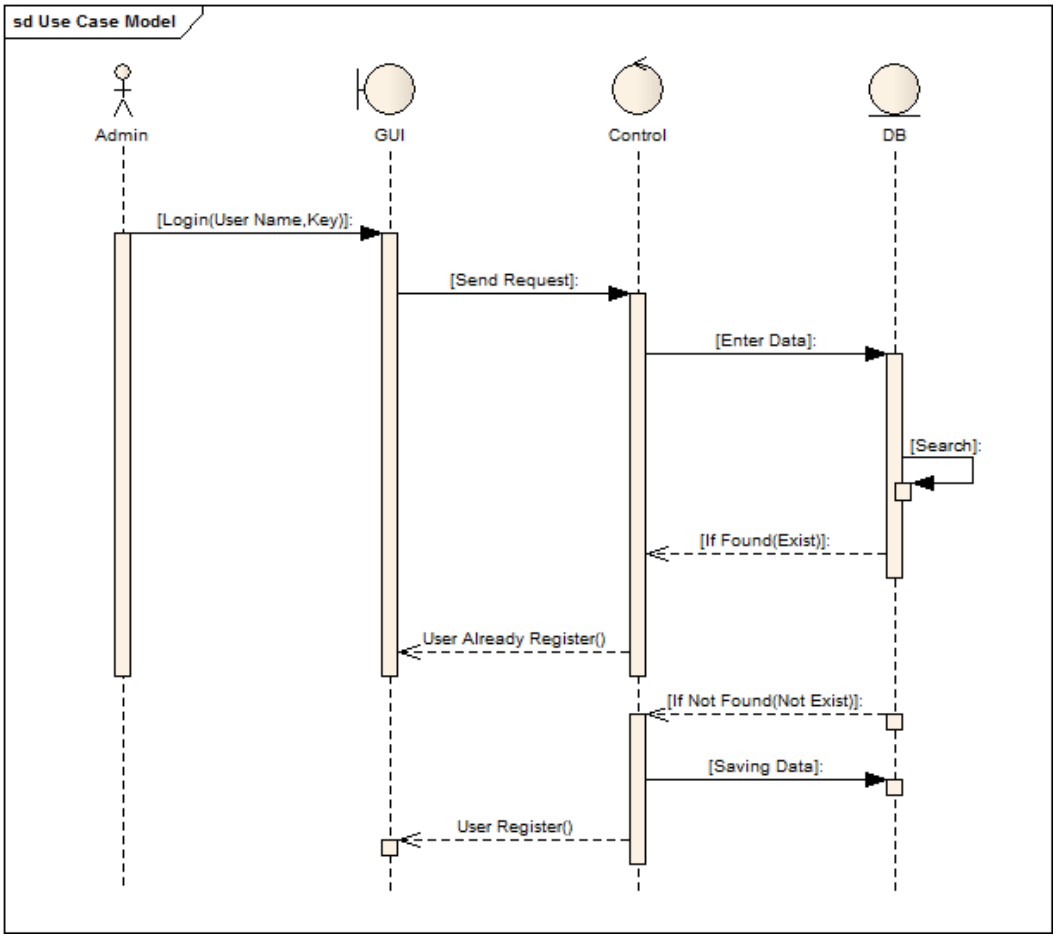
هو عبارة عن مخطط زمني يبين تسلسل حدوث عملية معينة والرسائل التي تنتقل بين مجموعة من الكائنات.[11]

تمثل عملية الدخول للنظام أول عملية يقوم بها المُستخدم، حيث يقوم بإدخال اسم المُستخدم وكلمة المرور، فيتم التأكد من صحة البيانات المدخلة بمقارنتها مع الرقم التسلسلي المخزن بقاعدة البيانات الخاصة بالنظام باستخدام دالة التأكد (Checker)، فإذا كانت البيانات غير صحيحة يتم إيقاف المُستخدم، وإذا كانت صحيحة يسمح للمستخدم بالدخول للنظام.



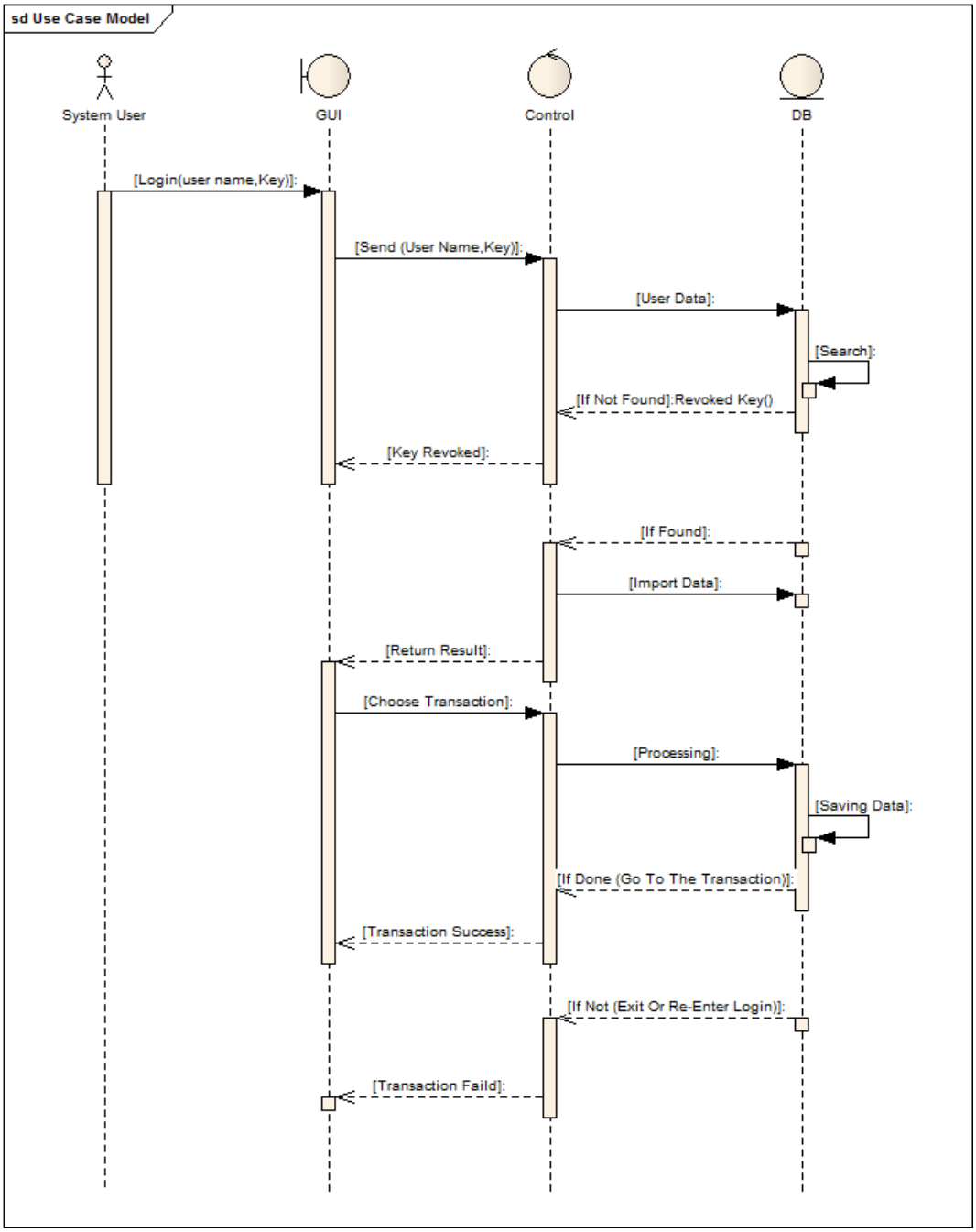
الشكل(2.6) : عملية الدخول للنظام

في هذه العملية يقوم المدير بإضافة عميل، وذلك بإدخال بيانات العميل وإضافة شهادته الرقمية بعدها تخزين اسم العميل وكلمة المرور الخاصة به المرتبطة بالرقم التسلسلي الموجود بالشهادة.



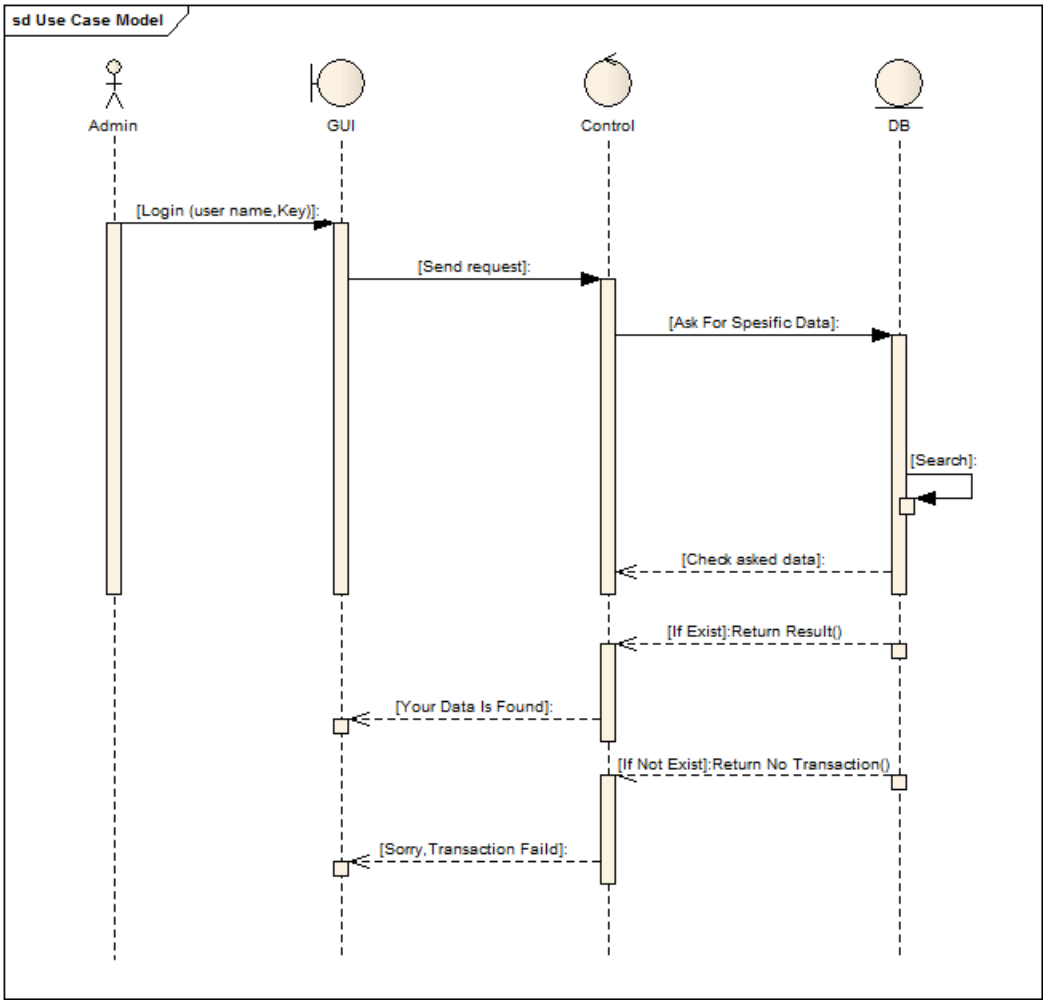
الشكل (3.6) : عملية إضافة عميل جديد

يقوم العميل بإدخال بيانات العملية التي يريد إجراءها، ويتم التأكد من صِحتها بعد ذلك تُرسل البيانات إلى قاعدة بيانات البنك ليتم تنفيذها وإعطاء إشعار بذلك.



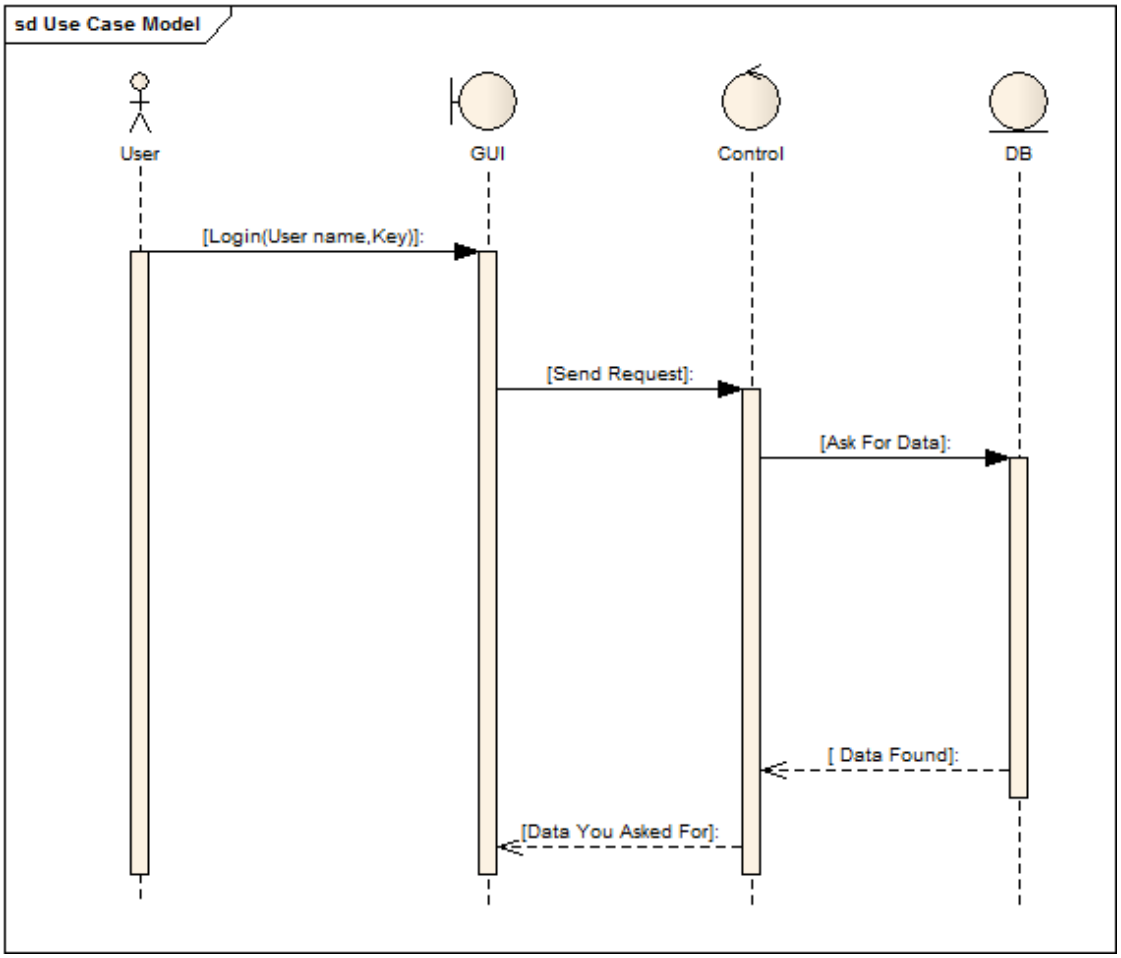
الشكل (4.6) : إجراء معاملة مصرفية

في هذه العملية يتم تأكد العميل من بيانات إحدى المعاملات المصرفية التي تمت، ويتم ذلك بإدخال بيانات محددة ومقارنتها بالبيانات الموجودة بقاعدة البيانات فإذا كانت موجودة يتم إرسال إشعار بوجود البيانات، وإذا كانت غير موجودة يتم إرسال إشعار بعدم وجود بيانات المعاملة المختبرة.



الشكل (5.6) : التأكد من إجراء معاملة مصرفية معينة

يتم في هذه العملية طلب إستعلام عن المعاملات المصرفية التي تمت، فيتم إرجاع تقارير عن جميع المعلاملات المصرفية التي تمت.



الشكل (6.6) : عملية إستخراج التقارير للمعاملات التي تمت

6.2.6 مخطط النشاط : Activity Diagram

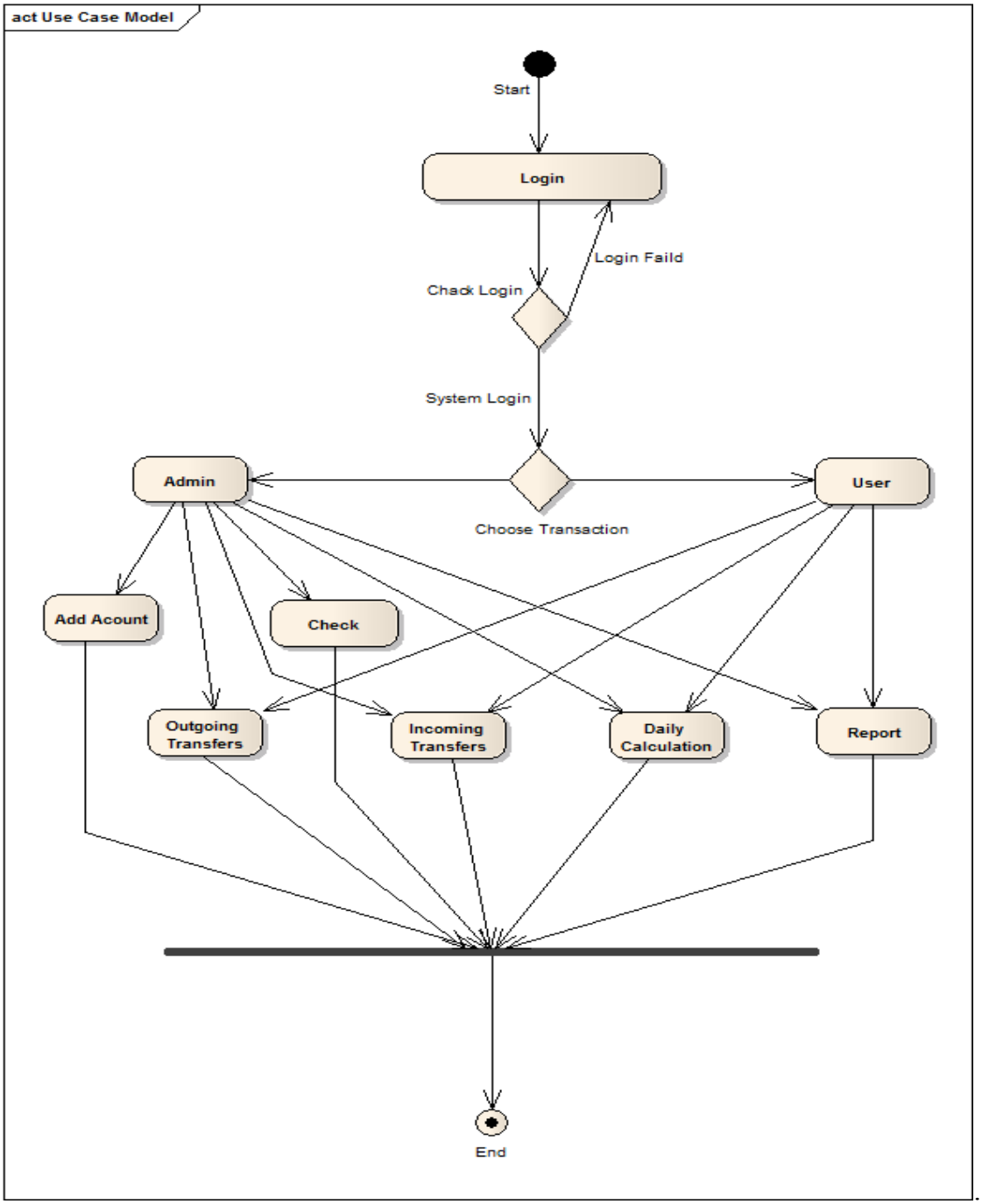
هو عبارة عن عمليات متتابعة مستندة على شروط معينة [11].

يعتمد نظام البنك على عدة عمليات متسلسلة تبدأ بعملية إدخال الإسم وكلمة المرور والتحقق منها، وبناءاً على ذلك يتم إدخال المُستخدم على حسب الصلاحيات المُخولة له للقيام بها :

- إذا كان مُدير النظام فإن العمليات التي يستطيع القيام بها كالاتى :
- إنشاء شهادة رقمية.
- التوقيع على الشهادة.

- إعادة التوقيع على الشهادة.
- إلغاء الشهادة.
- إضافة عميل.
- إجراء معاملات مالية.
- التحقق من المعاملات المالية.
- إصدار التقارير.
- إذا كان عميل لدى هذه الصرافة :

يقوم العميل بإدخال الشهادة الرقمية التي يمتلكها فإذا كانت الشهادة صحيحة، يُمكن له أن يختار العملية التي يريد إجراءها.



الشكل (7.6) : عمليات نظام الصرافة