**MODULE CI143: Requirements Project**

**Earl Aaron Balba**

**Owen Baldwin**

**Tom Grennan**

**Jake Thomson**

**Dan Thurston**

**Greg Stroud**

**John Vos**

**INDEX:**

1. [Stakeholder onion model](#ExecutiveSummary).
2. [Goal model](#GoalModel).
3. [Business process model](#BusinessProcessModel).
4. [Requirements table](#RequirementsTable).
5. [Requirements specification table](#RequirementsSpecificationTable).
6. [References.](#References)
7. [Appendix A.](#AppendixA)
8. [Appendix B.](#AppendixB)

**Stakeholder Onion Model:**

A stakeholder is an individual, department or organisation that have an invested interested in the product or who are directly affected by the product. They can be internal to the company that has been hired to deliver the project, such as; CEO’s, managing directors, project managers, programmers, the IT team etc… Furthermore, they can also be external, such as; the end users and customers of the software. In addition, we can further break them down into 3 levels, the first includes stakeholders who are actively involved in the project, the second are affected by work performed or the finished results and the third includes individuals, departments or organisations that exert a degree of influence over the project or deliverables[1](#IdentifyingStakeholders).

By determining who the stakeholders are we can get a clearer picture of who has which job and is affected by the end product. By doing so, we will get a clearer line of communication between everyone as we know what their job is. Furthermore, by doing so we can make sure the end product that is delivered is in fact the product that we set out to make to begin with. In addition, by doing so we can prioritise the stakeholders as some stakeholders are more important than others, such as, the executives, senior managers and customers. As they are the ones that are the most effected by the product. Furthermore, this is because they will all have different needs and wants. For example, the customer will want a straight forward and easy to use system, whereas the executives will be more interested in income, cost and work load.

One way of achieving this is by creating what is called a stakeholder onion model, by doing so, we create a relatively simple and straight forward visual representation. Furthermore[2](#StrategicRequirementsAnalysis), the first layer represents the stakeholders that are closely involved with the creation of the product (e.g. the project manager). The second layer includes stakeholders whose work changes once the solution is defined (e.g. end users). The third layer includes the sponsors, executives and subject matter experts who have invested interest in the end product / system. The fourth and final layer includes the external stakeholders, such as, the customer and suppliers.

**Wholesalers**

**US Attorney General**

**Customer**

**Nicole Keman**

**Transport Company**

**ATM Machine Manufacturers**

**Bank**

**CEO (you)**

**Government**

**Goal Model:**

When starting a new project, it is extremely important to set detailed goals and a vision of what it is exactly you want to achieve. By doing so, you can determine the scope of the technical and organisational components of the project; furthermore, this will give you an idea of how many resources you’re going to have to (and therefore willing to) allocate to the entire project and establish clear deadlines & the expected results. For example, if from the beginning you set a clear and detailed plan of the objectives, you are creating a solid foundation on which the project can be created and leaving little to no tasks open to assumptions from the people working on the project. Therefore, achieving the goals that were set out and not what an employee interpreted it to be.

Secondly, by creating a goal model you can foresee some of the obstacles you will come across and can plan ahead on how to overcome or avoid these obstacles. In addition, you can change some of the goals slightly to allow for changes that may occur further down the line, as well as, giving everyone clear guidance as to what it is that is expected from them.

When creating the goal model, we decided that the main vision of the company is to create a legal banking service for companies selling weed that would in turn indemnify the companies from being prosecuted by the government. Therefore, the vision being to “Create a legal banking service for companies selling weed that will indemnify them from prosecution”. In this vision however is also a goal, to indemnify companies from prosecution from the government, hence goal 1. Furthermore, a major part of the banking side was the ATM’s side of the banking, therefore, we create a separate goal for each part of the ATM process to lay out what it is the company wanted from the ATM’s. Next, we looked at what the armoured trucks tasks were and created 2 goals for them, linking directly into the ATM as the goals for the company wanted the ATM’s to do would directly impact what the security team would do. Finally, we looked at what we were hired to do and what our goals would be and how they would be affected by the banking system as a whole.

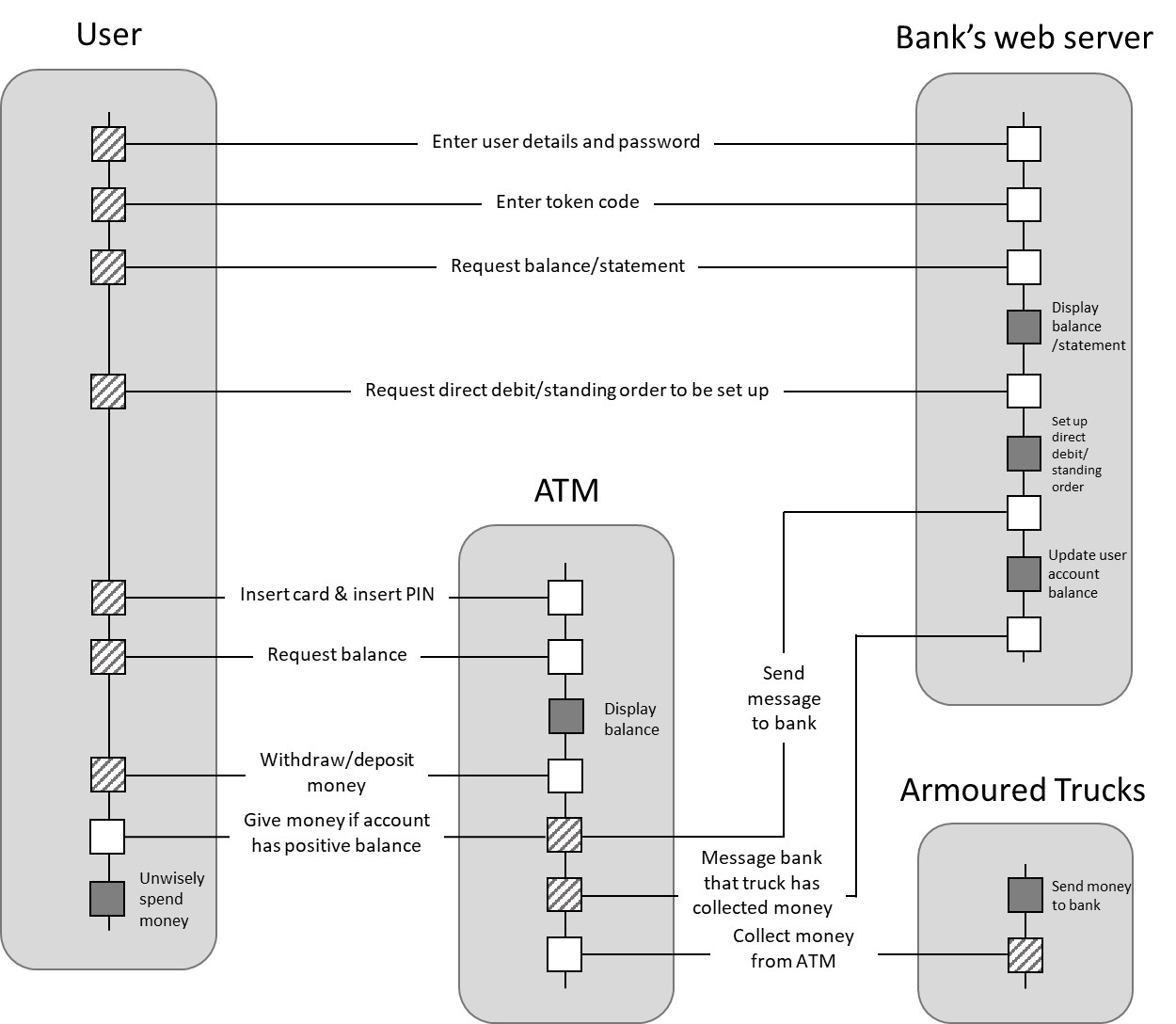
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **V: Create a legal banking service for companies selling weed that will indemnify them from prosecution.** | | | | | | |
| G1: Indemnify companies from prosecution from the government. |  | G2: ATM’s installed on the premises of the stores. |  | G4: Set up regular collections of cash from the ATM’s by the armoured trucks. |  | G3: Create an online banking service. |
|  |  |  |  |  |  |  |
|  |  | G2.1: Accept Deposits. |  |  |  | G3.1: Financial record system. |
|  |  |  |  |  |  |  |
|  |  | G2.2: Pay out cash |  |  |  | G3.2: Balance enquiry & statements. |
|  |  |  |  |  |  |  |
|  |  | G2.3: update customers’ accounts immediately. |  |  |  | G3.3: Receive customers Balance. |
|  |  |  |  |  |  |  |
|  |  | G2.4: Offer balance enquiries, cash deposits & cash withdrawal. |  | G4.1: ATM’s must be able to send a message to the bank every time they accept money, give out money or are unloaded by the security vans. |  | G3.4: Receive messages from remote ATM’s. |
|  |  |  |  |  |  |  |
|  |  | G2.4: Refuse to give out money if customers whose account is below 0. |  |  |  | G3.5: Standard internet banking facilities. |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  | G3.5.1: Ability to set up direct debits & standing orders |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  | G4: Easy to use & secure. |
|  |  |  |  |  |  |  |
| Index:  V: abbreviation for vision.  G: abbreviation for goal |  |  |  |  |  | G9: User must have correct password & enter token number. |

**Business process model:**

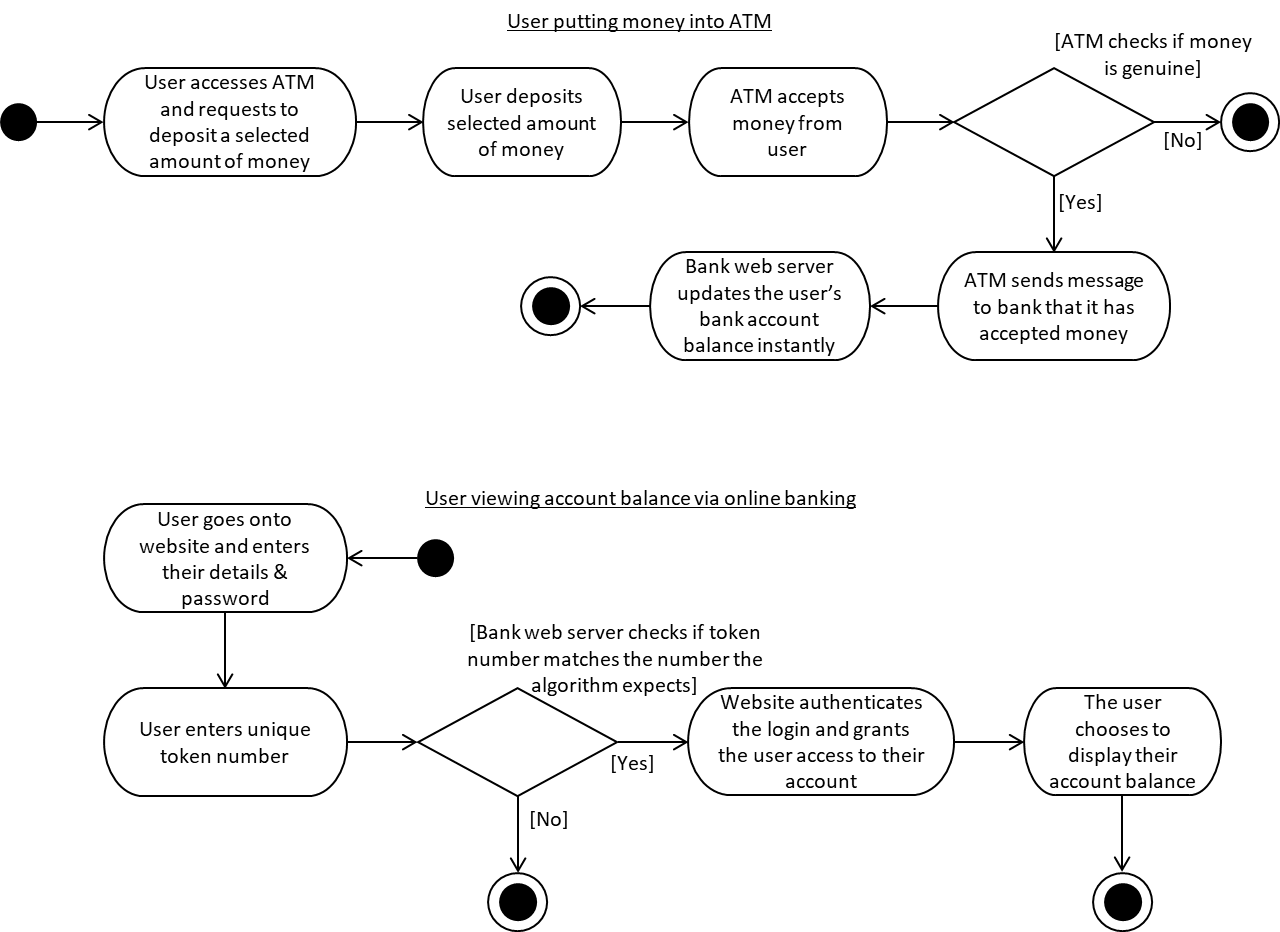
Business process modelling is used by business analysts to visually lay out an organization’s current business processes. It allows them to see what the projects current processes are, how they’re being carried out and how to potentially improve them. Furthermore, a business process model allows an organisation to illustrate the project’s process visually/digitally, enabling them to be able to present flaws/improvements in a very easy manner to other people working on the project. Business process modelling can be illustrated in a number of ways, one of the most common would be using Business Process Modelling and Notation (BPMN).

We have decided to create three separate business process models altogether, one (figure #) to show the general process of the whole system, one (figure #) to show the process of a user taking money out of an ATM, and one (figure #) to show the user accessing their account via online banking. We have decided to illustrate the first process model as a role activity diagram, this allows us to see what each of the roles in the system do, and how they all link together. As you can see from the first diagram, figure #, there are four parts to the whole system, the user, the ATM’s, the online web service and the armoured trucks, a quick glance will reveal that the user should be able to interact with both the ATM’s and the bank’s web server, however they have no interaction with the armoured trucks at all. These types of information can only be shown in diagrams such as role activity diagrams, and you could not display said information in diagrams such as UML activity diagrams for example.

Our last two business process diagrams, figures # & #, show the process a user goes through to use certain features of the system. We’ve decided to illustrate these as UML activity diagrams; UML standing for Unified Modelling Language. This decision was made as UML is a widely acknowledged language that is easily understood by IT and business managers. It can easily show the path the system goes through to complete certain tasks, what decisions it makes throughout, and the outcomes of those decisions. As seen in figure #, when a user wants to put money into an ATM, you can see the ATM only has a few certain tasks that it needs to carry out, accepting the money, checking the legitimacy of the money being deposited, rejecting the transaction if money is fake, and sending a message to the bank if everything has gone through.

**Role Activity Diagram**

**UML Activity Diagram**



**Requirements table:**

The aim of the requirements table is to clearly layout the key elements that the business process or system must fulfil. The information in the first column acts as an identifier for each requirement so that they can be referenced in other documentation. The following column is a full description of the requirement. This is where the requirement itself is clearly defined and specified. Following this is a column which states the type of requirement. There are four different types of requirement and they are defined as follows:

Business -

Functional –

Non-Functional -

User –

The type of requirement also denotes the first letter of the ID (i.e. the first functional requirement would be FR1 and the third business requirement would be BR3).

The fourth column states the owner of the requirement which is the person whom is in charge of ensuring that the requirement is fulfilled and will have control over making sure it is implemented correctly.

The final column states the importance of the requirement. It informs the reader whether it is absolutely vital to the functioning of the system (*must have*), whether it is just a preferred option (*should/could have*) or even something that was considered but ultimately not feasible (*won't have*).

**Requirements table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reqt ID | Requirement | Type | Owner | Priority |
| FR1 | User must insert a credit/debit card and enter a unique PIN to access their account (ATM). | User | Customer | Must have |
| BR1 | Indemnify companies from prosecution from the government. | Business | CLO | Must have |
| BR2 | ATM’s installed on the premises of the stores. | Business | Engineering manager | Must have |
| BR3 | Set up regular collections of cash from the ATM’s by the armoured trucks. | Functional | Transport manager | Must have |
| FR2 | ATM's must accept cash deposits from user | Functional | ATM Engineering manager | Must have |
| FR3 | ATM's must pay out cash when user withdraws from account. Card is returned to user and is logged out at the end of transaction. | Functional | CIO | Must Have |
| FR4 | ATM's must communicate with bank server to update customers’ accounts after a transaction has been made | Functional | CIO | Must have |
| QR1 | Within 25 seconds. | Functional | Customer | Should have |
| UR1 | After the customer selects the balance enquiry option, their account balance needs to be displayed on the screen. | Functional | CIO | Must have |
| FR5 | Bank must be notified when ATM funds are above or below a certain amount. | Functional | CIO | Could have |
| FR6 | The ATM must check whether the user has enough funds in his account. If the amount the user requests to withdraw results in their account balance dropping below £0, the ATM must deny the transaction, inform the user and not release any funds. | Functional | CIO | Must have |
| FR7 | ATM must notify bank of every transaction. | Functional | CIO | Must have |
| FR8 | ATM must notify bank when unloaded by security vans. | Functional | CIO | Must have |
| QR2 | ATM must verify notes are genuine. | Non-functional | Engineering manager | Must have |
| BR3 | Create an online banking service. | Business | CIO | Must have |
| FR9 | The online banking software must allow the user to set up a standing order to a payee's bank account for a specific date. | Functional | CIO | Must have |
| FR10 | The online banking software must allow external companies to arrange for regular direct debit payments from a payer's account to be made to the company account on a specific date. The payer needs to have accepted and confirmed the direct debit before the payment is setup. | Functional | CIO | Should have |
| FR11 | On the date specified, the bank sends the funds from the user's bank account to the payee's bank account according to the standing order or direct debit already in place. The payment is given an ID number so details of the payment can be later found and linked to (for example, an account statement) | Functional | CIO | Must Have |
| UR2 | The online banking service must allow users to view their account balance online | Functional | CIO | Must have |
| UR3 | The user must be able to view account statements showing all financial transactions between specific dates. | User | CIO | Should Have |
| FR12 | The online banking service must verify the user using a password and unique token codes (number generator) to allow the user access to their account | Functional | Customer | Must have |
| FR13 | Token device should generate unique number when activated. When this code submitted to the bank server via the website, the bank server is able to identify the user from their unique token code, along with other credentials | Non-functional | Engineering manager | Should have |

**Requirements specification table:**

Requirement specifications describe the functionality of a system’s software and provide instructions to aid in the development process[3](#RequirementsSpecificationTableRef). Often these specifications are documented in a large, detailed textual document, which can be misinterpreted due the size and complexity. Displaying the information in a table provides the benefit of having all information for specific requirements in one place, and illustrate point with visual aids such as diagrams and models to add clarity to the requirement.

The following table provides the key information for each requirement. Only functional and user requirements are included in specification requirements table, as these are the most relevant for the software development process.

The nine columns in this table are:

1. Reqt ID: Each requirement has a unique ID to identify the requirement
2. Requirement: A description of the requirement
3. Data: All data required for the process/requirement, this will often overlap with input and output
4. Process: A description of how the requirement is performed as a process from start to finish. Many of the requirement processes below reference data models attached to this document.
5. Inputs: Data that is passed into the software, this can be inputted by a user or from another process, for example, the user entering their PIN in an ATM.
6. Outputs: The resulting data involved in a process, for example if a customer requests to view their account balance, the output would be the system displaying the account balance on the screen.
7. Dependencies: Any other requirements which must happen before, or which may affect the process of the specific requirement, for example, the customer must be logged on to view their account balance.
8. Quality: The quality (non-functional) requirements which are relevant to the specific requirement, such as speed, reliability, usability.
9. Priority: The priority assigned to the requirement from the previous table seen in section 4.

Take for example FR7, first identifies the requirement with the ID. The requirement stated the ATM must notify the bank after every transaction. The data involved contains details about the account, the withdrawal, the address where the information must be sent and the time and date of the transaction. The process is described, and two data models are referenced for further instruction. The input in this case is the amount of cash withdrawn or deposited. The output is information sent to the bank server: amount withdrawn or deposited, where it is being sent, the account number of the user making the transaction etc. FR7 does have dependencies, it is dependent on there being a transaction (FR2 or FR3) for this information to be sent. A quality associated with this process is reliability, as the data must be correct otherwise the wrong amount of funds could be added or taken from the user’s account. Another quality which is important for this process is speed, as if this information is not updated quickly, their account could show more or less funds than the user actually has in their account. Finally, the priority is which was previously assigned when creating the requirement.

**Requirements Specifications Table:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reqt ID: | Requirement: | Data: | Process: | Inputs: | Outputs: | Dependencies: | Quality: | Priority: |
| FR1 | User must insert a credit/debit card and enter a unique PIN to access their account (ATM). | Credit/debit card number, PIN, access granted (yes/no). | See FR1 (login process) Scenario in [Appendix A](#AppendixB) to this document. See [Appendix B](#AppendixA) for ATM UI. | Credit/debit card,  PIN, confirmation details are correct (yes/no). | Credit/debit card number, PIN. |  | Reliability: connects the link between card holder and their account 100% of all cases when details are correctly entered  Usability: the system must be easy to use, navigate and intuitive for the user  Speed: credentials must be verified within 5 seconds of the user inputs their password. | Must have. |
| FR2 | ATM's must accept cash deposits from user. | Cash counted, confirmation amount is correct (yes/no), account number, transaction ID, message to user. | See [FR2](#FR2) (deposit process) Scenario in [Appendix A](#AppendixA) to this document. See [Appendix B](#AppendixB) for ATM UI. | Cash, confirmation amount is correct (yes/no). | Confirmation message or declined message, confirmed amount, account number, cash returned. | FR1 must occur before FR2 proceeds | Accuracy: cash inserted by the user is counted correct 99% of all cases.  Usability: the system must be easy to use, navigate and intuitive, including prompts for when and where to enter their cash.  Speed: cash must be counted within 10 seconds after being accepted by the ATM | Must have. |
| FR3 | ATM's must pay out cash on cash withdrawals. Card is returned to user and is logged out at the end of transaction. | Amount of cash requested, account balance, message to user, amount of cash released, transaction ID, card returned to user, log out. | See [FR3](#FR3) (withdrawal process) Scenario in Appendix A to this document. See [Appendix B](#AppendixB) for ATM UI. | Amount of cash requested (by user), account balance (from bank server). | Message to user, amount of cash released, card returned to user, transaction ID, log out. | FR1 must occur before FR3 proceeds.  FR6 must occur before FR3 proceeds and may affect outcome. | Accuracy: the cash released from the machine must be the exact amount requested by the user (and approved) 100% of all cases  Usability: the system must be easy to use, navigate and intuitive. | Must have. |
| FR4 | After the customer selects the balance enquiry option, their account balance needs to be displayed on the screen. | Account number, account balance. | See [FR4](#FR4) (balance enquiry process) Scenario in Appendix A to this document. See [Appendix B](#AppendixB) for ATM UI. | Display account balance option selected. | Account number, account balance. | FR1 must occur before FR4 proceeds. | Reliability: correct account details of the card holder are retrieved 100% of all cases.  Usability: the system easy to user, navigate and intuitive.  Speed: account balance must be displayed within 5 seconds of the request. | Must have. |
| FR5 | Bank must be notified when ATM funds are above or below a certain amount. | ATM number, location, balance,  IP address of bank server, threshold amount exceeded. | The ATM sends a message informing the bank that the contents are above or below a certain threshold. | Threshold amount exceeded. | ATM number,  Location, IP address of bank server  Message:  ATM contents are almost full, **or** ATM contents are almost empty |  | Reliability: the details of the ATM (location, number etc.) are correct in 99% of cases. | Could have. |
| FR6 | The ATM must check whether the user has enough funds in their account and If the amount the user requests to withdraw results in their account balance dropping below £0, the ATM must deny the transaction, inform the user and not release any funds. | Account number, account balance, withdrawal amount. | See [FR6](#FR6) (withdrawal process) Scenario in Appendix A to this document. | Amount user wishes to withdraw, account balance (from bank server). | Account number, card returned to user,  Message: Transaction is denied. |  | Reliability: correct account details of the card holder are retrieved 100% of all cases.  Speed: decision whether to pay out cash or reject the transaction must be made within 5 seconds after request has been made. | Must have. |
| FR7 | ATM must notify bank of every transaction. | Account number, amount withdrawn, amount deposited), IP address of bank server, time, date. | The ATM sends a message to the bank server whenever a user has made a transaction (withdrawal or deposit) containing details of the transaction.  See [FR7](#FR7) (deposit **and** withdrawal process) Scenarios in Appendix A to this document. | Amount withdrawn, amount deposited. | Amount withdraw, amount deposited, account number, IP address of bank server, time, date. | FR2 or FR3 must occur before FR8 proceeds. | Reliability: correct details of each transaction in 99% of all cases.  Speed: transaction notifications must be sent to the bank server within 1 second once the transaction has been authorised. | Must have. |
|  |  |  |  |  |  |  |  |  |
| FR8 | ATM must notify bank when unloaded by security vans. | Unique ATM number, ATM location, date, time, driver ID number, amount of cash unloaded from ATM, code to unlock ATM. | When the ATM is unloaded with a unique ATM code, a log is updated, and a message is sent to the bank server containing details of the ATM, driver and amount of cash unloaded.  See [FR8](#FR8) (ATM unloading process) Scenario in Appendix A to this document. | Driver ID number, unique code used to unlock ATM. | ATM number, location, time, date, amount of cash unloaded, driver ID number. |  | The details of the ATM (location, number etc.) and driver details are correct in 99% of all cases. | Must have. |
| FR9 | The online banking software must allow the user to set up a standing order to a payee's bank account for a specific date. | User bank account number, payee's details (bank account number, sort code, first name, surname, reference (if applicable)), date, time, amount of cash to transfer. | See [FR9](#FR9) (standing order process) Scenario in Appendix A to this document. See [Appendix B](#AppendixB) for Online Banking UI. | Payee's account number, date, payee's bank details (account number, sort code, first name, surname, reference if applicable), amount of cash to transfer. | Amount of cash to transfer, payee's account number, reference (if applicable). | FR12 must occur before FR9 proceeds. | Reliability: The link between payer and payee must be correct in 100% of all cases.  Usability: the system must be easy to use, navigate and provide detailed instructions. | Must have. |
| FR10 | The online banking software must allow external companies to arrange for regular direct debit payments from a payer's account to be made to the company account on a specific date. The payer needs to have accepted and confirmed the direct debit before the payment is setup. | Company bank details (name, account number, sort code), payer's bank account details (first name, surname, account number, sort code), amount to transfer (could vary each payment), date, time. | See [FR10](#FR10) (direct debit process) Scenario in Appendix A to this document.  See [Appendix B](#AppendixB) for Online Banking UI. | Payer's account details (first name, surname, account number, sort code), amount to transfer, time, date. | Amount of cash to transfer, business/ organisations account details (name, account number, sort code). | FR12 must occur before FR10 proceeds. | Reliability: The link between payer and payee must be correct in 100% of all cases.  Usability: the system must be easy to use, navigate and provide detailed instructions. | FR10. |
| FR11 | On the date specified, the bank sends the funds from the user's bank account to the payee's bank account according to the standing order or direct debit already in place. The payment is given an ID number so details of the payment can be later found and linked to (for example, an account statement). | Standing order/direct debit details: Payer/Payee’s details (first name, surname, company name, account number, sort code), amount to transfer, date, time, amount transferred transaction ID. | See [FR11](#FR11) (standing order **and** direct debit process) Scenario in [Appendix A](#AppendixA) to this document. | Standing order / direct debit details. | Amount transferred, date, time, transaction ID. | FR9 or FR10 must occur before FR11 proceeds. | Reliability: The link between payer and payee accounts and details of the standing order/direct debit payment must be correct 100% of all cases. | Must have. |
| UR2 | The online banking service must allow users to view their account balance online. | Account number, account balance. | See [UR2](#UR2) (view balance) Scenario in Appendix A to this document. See [Appendix B](#AppendixB) for Online Banking UI. | Account number (to link user request to bank server). | Account number, account balance. | FR12 must occur before UR2 proceeds. | Reliability: correct account details of the user are displayed 100% of all cases  Usability: the system must be easy to use, navigate and intuitive  Speed: the account information must be sent within 5 seconds after the request has been received by the bank’s web server. | Must have. |
| UR3 | The user must be able to view account statements showing all financial transactions between specific dates. | Account number, transaction details (date, time, payee/payer, amount transferred). | See [UR3](#UR3) (display account statement) Scenario in Appendix A to this document. | Account number (to link user request to bank server). | Account number, transaction details (date, time, payee/payer, amount transferred). | FR12 must occur before UR3 proceeds.  FR2, FR3 and FR11 will affect the content produced by FR12. | Reliability: correct account details of the user are displayed 100% of all cases.  Usability: the system must be easy to use, navigate and intuitive.  Speed: the account information must be sent within 5 seconds after the request has been received by the bank’s web server. | Must have. |
| FR12 | The online banking service must verify the user using a password and unique token codes (number generator) to grant access to their account. | Username, password, unique token code (generated by device), confirmation that details match (yes/no). | See [FR12](#FR12) (login process) Scenario in [Appendix A](#AppendixA) to this document. See [Appendix B](#AppendixB) for Token Device UI. | Username, password, unique token code. | Confirmation that details match (yes/no). | FR13 must occur before FR12 proceeds. | Reliability: connects the link between user and their account 100% of all cases when details are correctly entered  Usability: the system must be easy to use, navigate and intuitive  Speed: credentials must be verified within 5 seconds of the user inputs their password | Must have |
| R13 | Token device should generate unique number when activated so that when this code is submitted to the bank server via the website, the bank server is able to identify the user from their unique token code, along with other credentials. | Token code, time, date. | When the device is activated (button is pressed) a code will be generated based on the current time. Within a short time period, the bank server will accept the code if entered to the bank server, granting access to the user account if the code, password and username match. See [Appendix B](#AppendixB) for Token Device UI. | Time and date (from activating the token device). | Token code. |  | Usability: the system must be easy to use, navigate and intuitive  Reliability: codes generated by device should match the code generated by the bank in 99% of all cases, allowing for slight degree of error if generated during a cross-over period  Robustness: device should be durable with components protected from a sealed plastic case, which can withstand drops from a height of 180cm. | Should have. |

**References:**

# Bibliography

Cox, K. A., 2015. *Strategic Requirements Analysis.* s.l.:s.n.

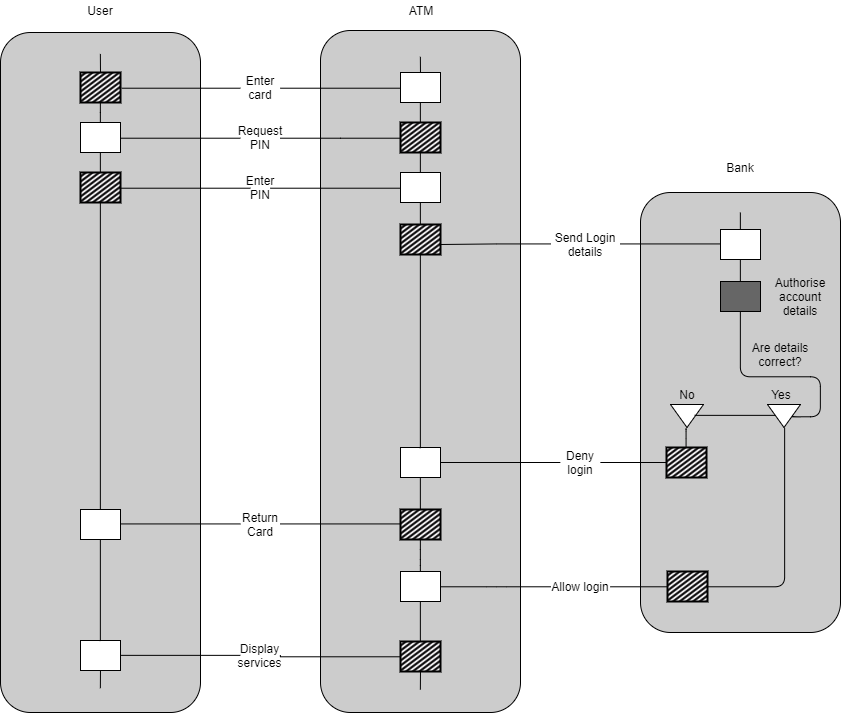
Levy, D., 2017. *Software Requirements Specification (SRS), What You NEED To Know.* [Online]   
Available at: http://www.gatherspace.com/static/software\_requirement\_specification.html  
[Accessed 6 December 2017].

Lohrey, J., 2009. *the importance of identifying stakeholders in a project.* [Online]   
Available at: http://smallbusiness.chron.com/importance-identifying-stakeholders-project-74730.html

Rouse, M., n.d. *techtarget.* [Online]   
Available at: http://whatis.techtarget.com/definition/business-process-modeling

# Appendix A:

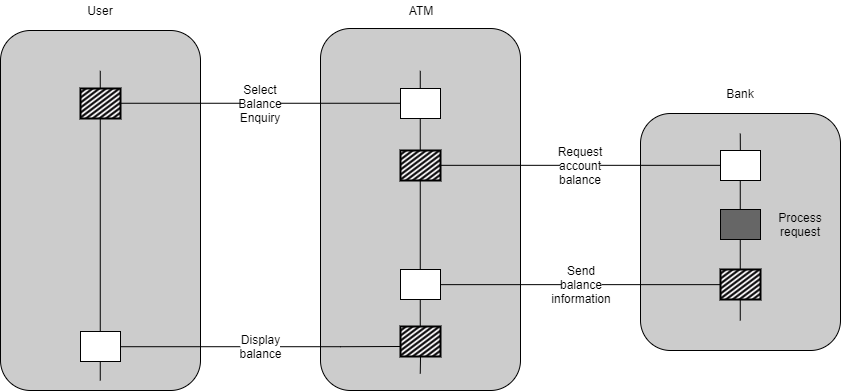
### FR1 - Login process



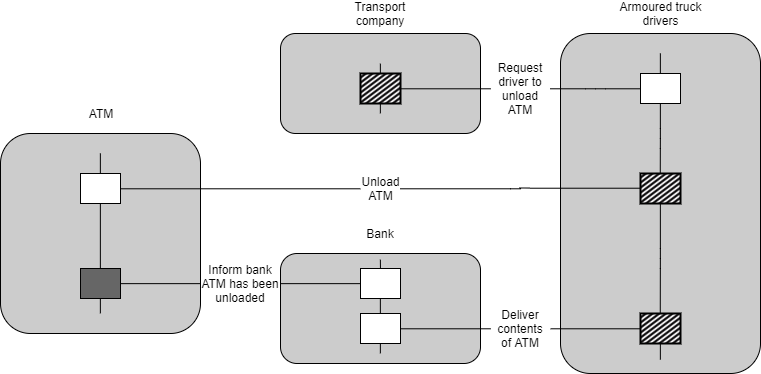
## FR2 - Deposit process

## FR3 and FR6 - Withdrawal process

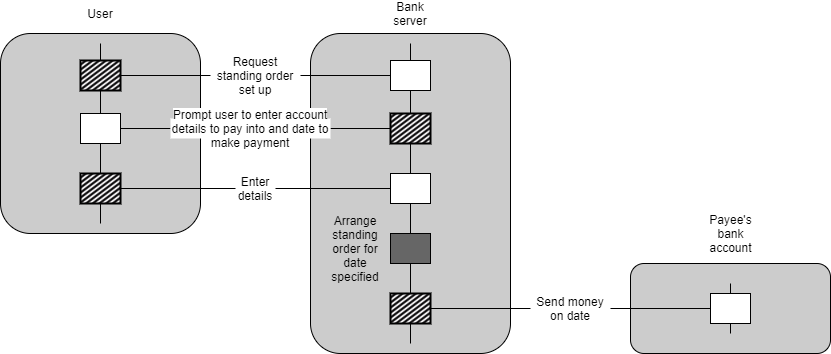
### FR4 - Balance enquiry process



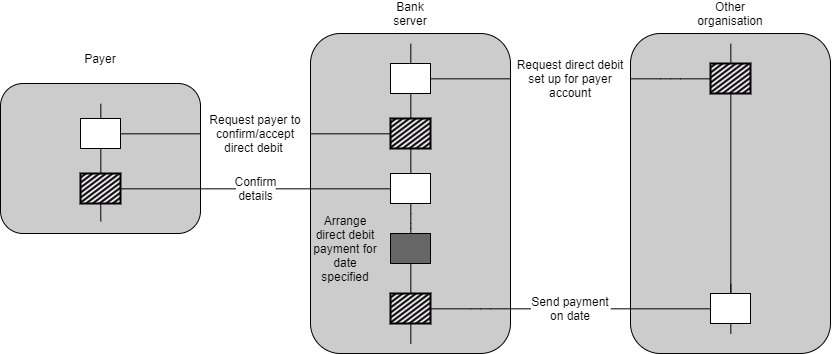
### FR8 – ATM unloading process



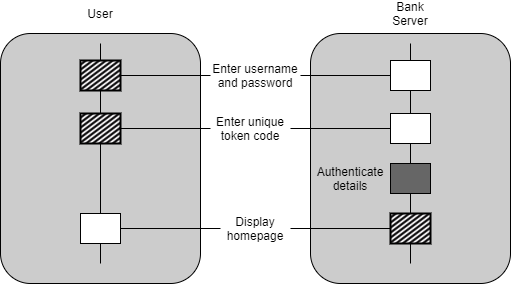
### FR9 and FR11 Standing order process



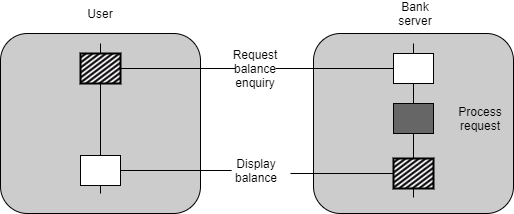
### FR10 and FR11 Direct debit process



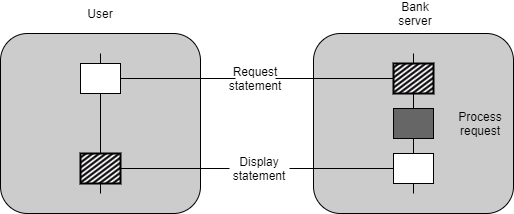
### FR12 – Login process



### UR2 – Display balance



### UR3 – Display account statement



# Appendix B:

**Online Banking**

**HOME**

**My Account**

**Transfers & Payments**

**Standing Orders**

**SERVICES**

**Set up a Standing Order**

**Manage Account**

**Make a Transfer**

**Help & Info**

**Settings**

**Contact Us**

***(Account # 3)***

***(£000.00)***

View transactions history

***(Account # 2)***

***(£000.00)***

View transactions history

**HOME**

**My Account**

**Transfers & Payments**

**Standing Orders**

**My Account**

***(User’s Full Name)***

*(Account Number)*

*(Sort Code)*

*(Card Number)*

***(Account # 1)***

***(£000.00)***

View transactions history

***£***

**HOME**

**My Account**

**Transfers & Payments**

**Standing Orders**

**Transfers & Payments**

**From:**

**To:**

**Ref:**

**Amount:**

*(e.g. Train tickets)*

**Or add a new recipient**

*(Please select an existing account)*

*(Please select an account)*

**Set up a new standing order**

***Standing Order #2***

Delete

Amend

***Standing Order #1***

Delete

Amend

**HOME**

**My Account**

**Transfers & Payments**

**Standing Orders**

**Standing Orders**

**ATM**

1

2

6

9

5

4

8

7

3

Insert card

1

2

6

9

5

4

8

7

3

Choose an option:

Withdraw Account balance

Deposit Exit

1

2

6

9

5

4

8

7

3

Withdraw: Select amount

30

5

40

10

500

15

Specified amount

20

1

2

6

9

5

4

8

7

3

Enter specified amount:

0:00

1

2

6

9

5

4

8

7

3

Are you sure this is the correct amount?

55:00

1

2

6

9

5

4

8

7

3

Printing amount. Please take your card.

**Token Device**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**0**