

Specfication Document

Project Name: Mobile App

Team Number 02

|  |  |  |
| --- | --- | --- |
|  |  |  |

Document Information

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
| Project Name: Lloyds Banking App |  | | |
| Prepared By: Team 2 |  | Preparation Date: |  |
| Email / Phone: |  |  |  |
| Document Version No: | 0.0 | Document Version Date: xx/xx/xx |  |
|  |  |  |  |

Version History

| Ver. No. | Ver. Date | Revised By | Description |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

## Purpose (Executive summary)

Purpose of the document, description of project task, system benefits, objectives and goals

**--------------------------------------------------------------------------------------------------------------------------**

**The Remainder of the document must contain:**

1. **Background & Analysis**
   1. **Analysis Process**

We researched into other banking apps to gain an idea of design and what possible features other banks have that Lloyds doesn’t. Then we listened to Lloyds’ research from young people and what they would like from an app (quick, clean, etc.). We also put our student brains together to think of what we would like as functionality in a banking app.

* 1. **Analysis**

The app would be designed for the student environment and aim to get them into banking earlier and banking better.

1. **Roles and Deliverables**
   1. **Roles**

Team Leader **– Matthew Selby (Student No. 130545637)**

Deputy Team Leader **– Josh Greenwood. (Student No. 130334835)**

The team will consist of several different roles. These roles are flexible; any person can take on one or more roles at any particular time, should it be necessary.

The following outlines what each of the roles will involve within this agile team:

**SCRUM Master – Matthew Selby**

Ensure that the agile process runs smoothly by protecting the team from disruptions and impediments.

Organise meetings including sprint planning, daily scrum (over a social networking website as it is not always feasible to do this face-to-face) and sprint retrospectives. Document the outcomes of these meetings.

To motivate and keep a high team morale.

Maintain high team efficiency and creativity.

**Product Owner – Joshua Greenwood**

Responsible for creating and maintaining the backlog

Prioritising the backlog

Ensures that quality work is produced by accepting or rejecting the work produced by the development team

**Development Team – All team members are in the development team**

The development team is made up of the system designers, UI designers, programmers, testers and documenters:

**System designers** will design the architecture of the system. They decide how the separate components of the system will work together. They will be directed by members of the group who have declared system design as a personal strength.

**UI designers** will design the look and feel of the system.

**Lead designer –** A Lead Designer will be decided by consensus on who fits the role the most, by the team after gaining some experience designing together. If the consensus is split, the project leader will make the tiebreaking decision.

**Programmers** will implement the system using the designs produced by the system designers. They will also integrate the UI design produced by the UI designers into the system.

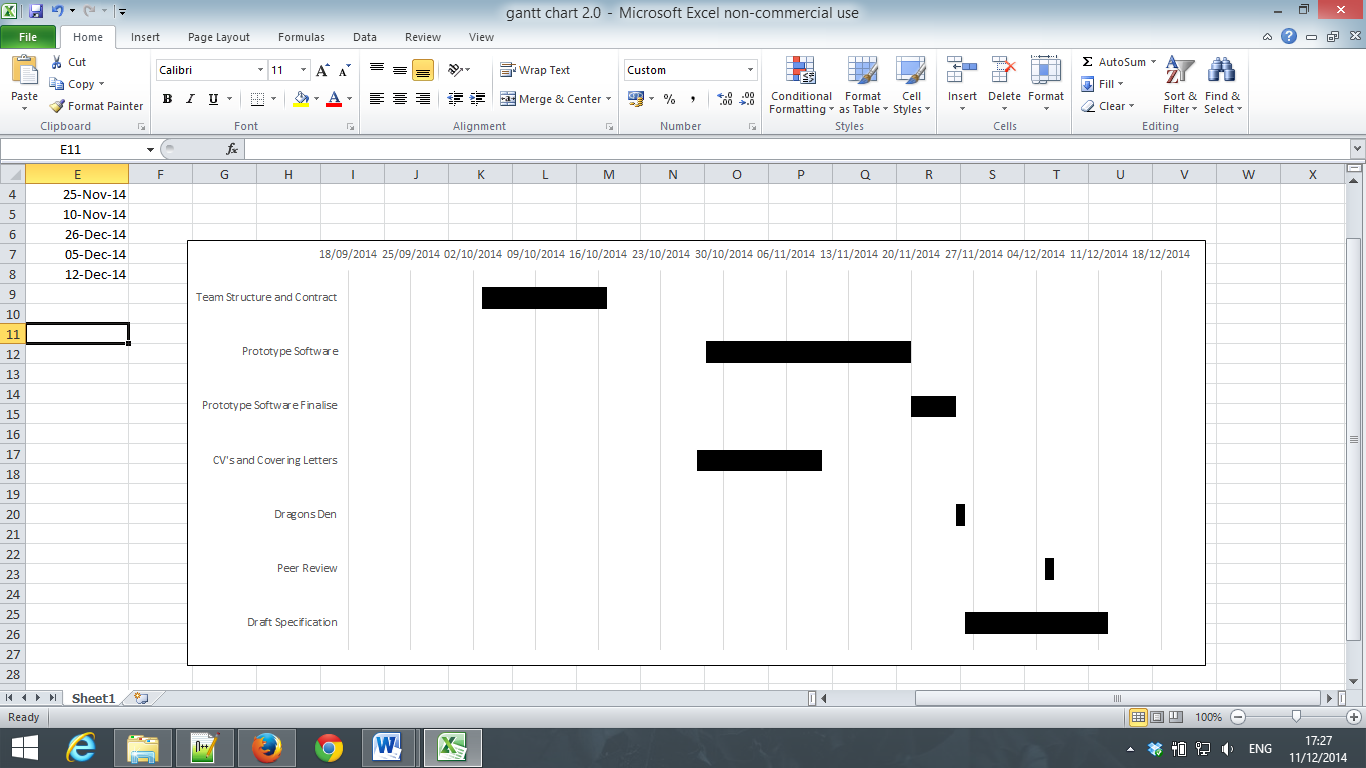
**Lead Programmer –** A Lead Programmer will be decided by consensus on who fits the role the most after gaining some experience programming together. If the consensus is split, the project leader will make the tiebreaking decision.

**Testers** will write and execute automated non-functional testing to ensure the system is behaving correctly and meets the requirements. Testing also includes installation, compatibility, regression, smoke and acceptance testing. All team members take part in manual, functional testing.

**Documenters** will document the user manual of the system. They will also document the system manual for the current programmers and any future programmers who join the team at a later time.

* 1. **Deliverables** – we are going to deliver a Lloyds bank android app for Students.

1. **Project Plan**



Gantt Chart for Semester 1.

1. **Hardware and software platforms to be used for developing and running your solution**

Software:

JustinMind Prototyper

JIRA

GIT

Hardware:

Android Mobiles

1. **References** – to all documents you use e.g. for research and also the initial Project Brief etc
2. **Definition of terms** – all terms used within the document e.g. Carbon Footprint – what is it? Brief description. This section can also be put at the end of the document if you think it best.
3. **Solution requirements**

For each requirement indicate its Priority e.g. High, Medium, Low. As you (your team) are the suppliers of the system you need to provide ECLS with information on what requirements you will complete within the development of the system. In the supplier compliance section your responses must be Yes, No or Partial coverage of the requirement by the solution you are proposing. If partial you need to specify why you will onlyu deliver part of the requirement in the Supplier Comment column e.g. if a requirement is high priority and you will not provide this you must give a short written rationale.

* 1. Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Priority  (H, M, L) | Supplier  Compliance  (Full, partial or will not be delivered) | Supplier  Comment |
| Makes Transfers | H | FULL | Basic requirement |
| Make transfers instantly and securely by touching contactless payment cards | L | N/a | n/a |
| Has a secure login | H | FULL | Basic requirement |
| Shows balances and statements | H | FULL | Basic requirement |
| Has a main screen to navigate the app. | H | FULL | Basic requirement |
| Main screen is engaging and uses well designed icons | M | FULL | Not difficult to implement |

* 1. Non-Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Priority  (H, M, L) | Supplier  Compliance (Full, partial or will not be delivered) | Supplier  Comment |
| Has an achievements based reward scheme for students | M | FULL | Moderate requirement, but an innovate feature we all want to see work well in the app. |
| Has a student banking planner. | M | FULL | Moderate requirement, but an innovate feature we all want to see work well in the app. |
| The achievement based rewards scheme is linked the users spending habits | M | N/a | N/a |
| The achievement based rewards scheme has an interesting map based design. | L | N/a | N/a |

NB: Compliance –Comments can relate to priority level and the reason for it e.g. we will only deliver part of this functionality at this time as it is low priority. Generally the team should aim to deliver all requirements that are noted as High (H) and Medium priority(M).

1. **Other considerations**
   1. Assumptions
   2. Constraints and Dependencies

**9. Software Design**

9.1 Overview of what the software will do and not do.

9.2 Modifications in light of comments made or changes deemed necessary

9.3 System Architecture – describe design decisions (Architecture diagram)

9.4 High level overview of how the functionality and responsibilities of the system are partitioned and assigned to components (deployment diagram, component diagram)

9.5 Package and Class diagrams which show dependencies between components

9.6 Show the dynamic behaviour of the system (sequence diagrams that shows communication between objects. Activity diagrams that show the state of objects).

9.7 GUI, Human Interface Views – Describe how the user accesses functionality

**NB:** Remember to associate aspects of design to the requirement number that they fulfill – communicate how your design meets the customer requirements.

**Guidelines**

* Only use diagrams where they will enhance explanation. Explain all diagrams that are used.
* Only design core parts of the system, not every small part of the functionality
* Always link between requirements and the design
* Any activity diagrams show link to the static structure of the diagram – (Class diagram)
* Do not submit any Javadoc with the document
* Page Limit – Max 30

**Marking Criteria**

* clearly communicated design decisions - is the approach adopted clearly stated and illustrated?
* Textual support for diagrams
* Format of the document according to the template
* Reasonable changes are clearly stated
* Presentation of major screen designs / screen shots
* Appropriate use of UML

Contribution Matrix for team work on the specification goes here.