|  |  |
| --- | --- |
| A picture of a winding road and trees  Library Management system  Assessment 1 | Abstract  Planning phase of the CS106 Library management system.  Ian Teves, Jack Giddens, Jordan Jenkins  CS106 |

Table of Contents

[Project Description (Pre-planning) 2](#_Toc117265798)

[Timeline & Constraints 2](#_Toc117265799)

[Project Constraints: 2](#_Toc117265800)

[Team Members & Responsibilities 2](#_Toc117265801)

[Software Process Model Used 2](#_Toc117265802)

[Tools Used 3](#_Toc117265803)

[Software Requirements (Planning) 4](#_Toc117265804)

[Identifying Functional & Non-functional Requirements 4](#_Toc117265805)

[Functional Requirements 4](#_Toc117265806)

[Non-functional Requirements 4](#_Toc117265807)

[Getting user requirements (User Testing) 4](#_Toc117265808)

[Interviewing (Open + Closed) 4](#_Toc117265809)

[Ethnographic research 4](#_Toc117265810)

[User Stories + Scenarios 4](#_Toc117265811)

[User Requirements 4](#_Toc117265812)

[Classifying/Organising gathered requirements 4](#_Toc117265813)

[Prioritising/negotiating requirements 4](#_Toc117265814)

[Requirement Specification document 4](#_Toc117265815)

[Requirement Validation by tutor 4](#_Toc117265816)

[UI Design/Prototype 5](#_Toc117265817)

[Sketches 5](#_Toc117265818)

[Lo-fi Frames 5](#_Toc117265819)

[Hi-fi Frames 5](#_Toc117265820)

[Screen Layout 5](#_Toc117265821)

[Main and Secondary Windows (Dialogue Boxes and pop-ups) 5](#_Toc117265822)

[Functions of each window/screen 5](#_Toc117265823)

[Form elements in each screen/window 5](#_Toc117265824)

[User-testing 5](#_Toc117265825)

[Presentation 5](#_Toc117265826)

# Project Description (Pre-planning)

## Timeline & Constraints



### Project Constraints:

1. Time – We have a large project to complete and submit within a short period.
2. Resources – We are to put it in few words, quite unfamiliar with the tools used for making the graphical interface.

## Team Members & Responsibilities

Ian Teves – Team Leader

Jordan Jenkins – Documenter & Programmer

Jack Giddens – Programmer

Due to time constraints, our team will be assuming any tasks that is left open to finish which is delegated by the Team Leader.

## Software Process Model Used

Due to time constraints our team will be approaching to finish the project using Agile Values modified to fit our teams process which makes use of sprints to finish tasks. This does mean that documentation will be at a minimum but will result in faster output of finished products for the client to make use of.

The Agile Manifesto that we will be following by order of importance are…

1. Daily communication is essential in a productive and functioning team
2. Simplicity – The art of maximising the amount of work not done is essential
3. Working Software is the primary measure of progress
4. Deliver Working Software Frequently, from a couple pf weeks to a couple of months, with a preference to the shorter timescale
5. Businesspeople and developers must work together daily throughout the project
6. The best architecture requirements and design emerge from self-organising teams

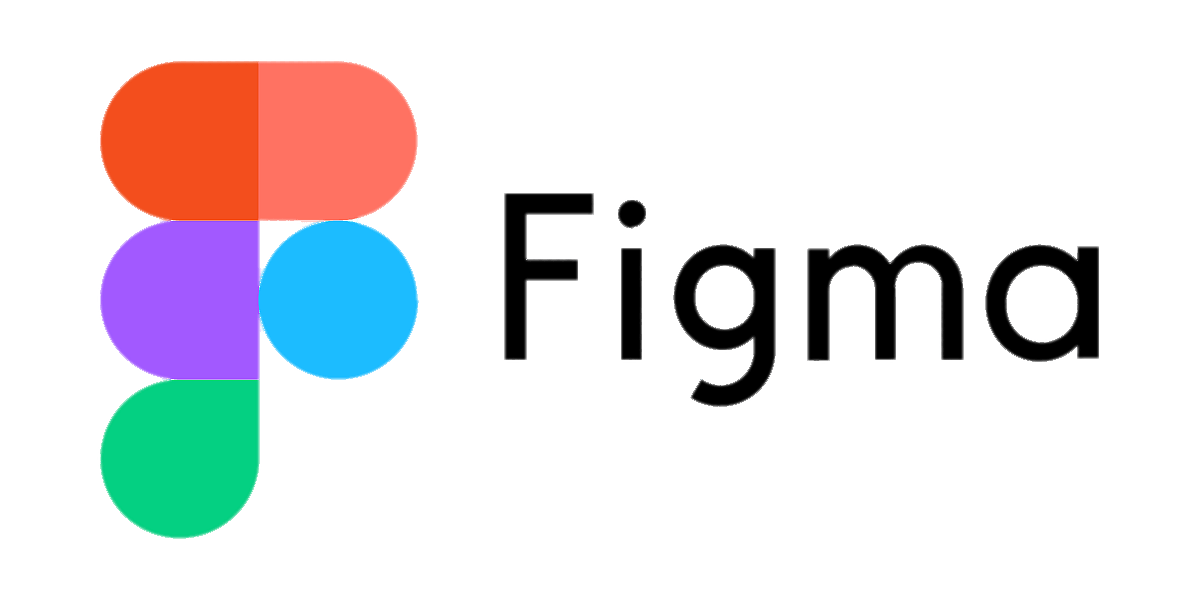
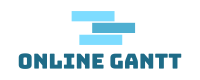
note… Team Leader will keep group in order by delegations of tasks

## Tools Used

To develop the library information system, we will need a few tools to get the job done. We will be dealing with a GUI, file managing, classes and objects and countless loops and algorithms for searching through data associated with each user.

* Visual Studio - Visual Studio is what will be used for writing backend code, such as dealing with files for storing user’s data working and working with objects.
* Logo

  Description automatically generated Qt Creator - Qt is the standard framework used in class for creating a GUI. It makes creating a GUI on a desktop or for mobile extremely straightforward and easy to edit if the design isn’t working out.
* A picture containing text, clipart, sign

  Description automatically generatedExcel - As we’ll need to store user data, one of the best ways to do it is using a CSV (comma-separated values) file. Excel can read and display in its cell format, which will make for much easier debugging.
* Figma – To create lo-fi and hi-fi prototypes, Figma will be used because of its simplicity to design, also its ability to recreate our ideas to the finest details.
* Trello – To create, delegate, and update tasks completed / to be completed
* Online Gantt – To create project timeline and track project completion rate / progress

# Software Requirements (Planning)

## Identifying Functional & Non-functional Requirements

A functional requirement defines the function(s) that the app should perform.

In the library app, for example, a user should be able to search for a title, and then the system takes the data input, searches the database, and should output any result that may be relevant.

Non-functional requirements describe how the app will deal with the work.

If multiple users were to use the library app at the same time, it should be able to maintain fast loading times.

### Functional Requirements

* Requirements go here…

### Non-functional Requirements

* Requirements go here…

## Getting user requirements (User Testing)

Include questions asked…

### Interviewing (Open + Closed)

Results go here…

### Ethnographic research

Observations during testing go here…

### User Stories + Scenarios

Basically, the users give feedback if something didn’t work how they’d expect, they come up with a few sentences illustrate how they want it to work.

### User Requirements

* Bullet point them here…

## Classifying/Organising gathered requirements

## Prioritising/negotiating requirements

## Requirement Specification document

* Use case Diagrams with explanation in case of functional requirements
* Detailing functional and non-functional requirements and assumptions

[How to Write a Software Requirements Specification (SRS Document) | Perforce](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document)

## Requirement Validation by tutor

# UI Design/Prototype

## Sketches

## Lo-fi Frames

## Hi-fi Frames

## Screen Layout

## Main and Secondary Windows (Dialogue Boxes and pop-ups)

## Functions of each window/screen

## Form elements in each screen/window

## User-testing

# Presentation