Project Plan

Project information	
Project	Cyber Library
Project Time-frame	15 weeks
Summary	Create a secure environment for teachers and students to share ideas, files, and assignments through mobile access and messaging.
Process impact: This plan will be used to evaluate and manage the project. Key assumptions that affect the plan should be documented here. The project plan should be updated throughout the life-time of the project.	

1- Summary of Project

Create a secure environment for teachers and students to share ideas, files, and assignments through mobile access and messaging.

2- Summary of Methodology

What general development approach will be used?

- Functionalities will be added one by one

- Subversion will be used to track changes to project source code and database schemas
- Programmers will each work using a personal test site, checking changes into subversion regularly
- The prototype site will be updated from subversion to pickup changes from developers, the user / testers are asked to test any new features
- The featureset is worked through according to a project plan in various development phases. At the end of each phase, the project is reviewed for changes to the features and the project plan is updated to reflect any changed goals

How will the project team be organized?

The development team will consist of 5 members

What development and collaboration tools will be used?

We plan to use the following tools extensively throughout the project:

- PyCharm IDE
- Github version control system
- Heroku hosting server

How will changes be controlled?

- After the feature complete milestone, new feature might be considered

- After the code complete milestone, new product source might be added
- All source code commit log messages must refer to a specific issue ID, after the feature complete milestone

How will this plan be updated?

This project plan will be updated as needed throughout the project. It will be placed under version control and every member can get this. Any change to the plan will cause an automatic notification to be sent to a project mailing list.

3- Work Breakdown Structure and Estimates

Step	Description	Estimate
1	Developer training	1 weeks
2	Inception	
2.1	Requirements gathering	2 days
2.2	Requirements specification	2 days
2.3	Requirements validation	2 days
3	Object design	
4	User interface design	
5	Database design	
6	Implement COMPONENT-NAME 1	
7	Implement COMPONENT-NAME 2	
8	..	
9	Implement COMPONENT-NAME n	
10	Testing	
11	Final presentation	
	Estimated time total	15 weeks

4- Deliverables in this release

Code name	Description	Date
Beta 1	A runnable application with at least 3 important use cases (function)	Week 6
Beta 2	All main use case are completed	Week 10
RC1	Deploy on host, open beta	Week 14
Final		Week 15

5- Risk Management

What are the main risks of this project?

- There are significant technical difficulties in building a web site and web application. This will be a risk because one person on our team has much experience with the relevant tools and technologies. Although the others will learn, we will certainly make some mistakes and suboptimal choices. We will address this risk by scoping the project such that we have enough time to train and to review the design and implementation.

- The schedule for this project is very short. We will manage this by planning a conservatively scoped functional core and series of functional enhancements that can be individually slipped to later releases if needed.
- The performance of the system will be significantly impacted by the decisions made during the database design task. None of our current team members has experience with database optimization. To address this, we will arrange a review meeting with an experienced DBA or hire a consultant from the database vendor.
- We could be underestimating known tasks. HOW TO AVOID/MITIGATE?
- We could be underestimating the impact of unknown tasks. HOW TO AVOID/MITIGATE?
- We could be underestimating the dependencies between tasks. HOW TO AVOID/MITIGATE?
- We could have misunderstood the customer's requirements. HOW TO AVOID/MITIGATE?
- The customer could change the requirements. HOW TO AVOID/MITIGATE?
- We could face major difficulties with the technology chosen for this project. HOW TO AVOID/MITIGATE?
- We could have low quality that demands significant rework. HOW TO AVOID/MITIGATE?
- We could incorrectly assess our progress until it is too late to react. HOW TO AVOID/MITIGATE?
- We could lose resources. E.g., team members could get sick, spend time on other projects, or quit. HOW TO AVOID/MITIGATE?
- There may be a mis-alignment of stakeholder goals or expectations. HOW TO AVOID/MITIGATE?

6- Project Planning Dependencies

Does this project conflict or complete for resources with any other project?

No, this is the only project what we are working on.

Are the same human or machine resources allocated to maintenance of past versions and/or planning of future versions during this release time period?

No, this is the first release and we will not plan the next release.

Dose this project depend on the success of any other project?

No, this project stands alone.

Are there any other important dependencies that will affect this project?

No, everything is covered above.