Software Project Management Plan

Flash Card Creation

March 4, 2019

Team Members
Joshua Folken
Melkam Getachew
Steve Thao
Vrusha Patel

Document Control

Change History

Revision	Change Date	Description of changes
V1.0	3/4/2019	Initial release

Document Storage
This document is stored in the project’s SVN repository at:
https://github.com/Mishti4812/Capstone

Document Owner
Group 6

Table of Contents

1 Overview

1.1 Purpose and Scope

Flash cards are a great to help students learn and retain information. You've been hired to create a digital substitute. The program should allow anyone to create an account and create sets of flashcards for studying. The creator of the flash card set can edit/add/delete from the set and share them with other users. When a user studies with a set of flash cards they will be randomly selected and allow the user to guess what is on the other side before revealing it. Stats of % are kept for history so the user can see their progress. Users can share flashcard sets. They can apply owner, copy and use privileges. Owner can add/edit/delete as well as use the cards normally. Copy rights allow people to copy the set as their own. The changes they make and own will not change the original set. Use privilege allow someone to run flash card quizzes with the set. Besides normal users, there are administrators that can disable accounts and perform site maintenance as needed. The choice of language and databases is up to you, however it must be a web solution that can be extended to mobile platforms.

1.2 Goals and Objectives

Project goals:

1. Establish the project on Mobile platforms
2. Easy usability by students to create and share as they wish
3. Make site able to be used on mobile through the web.
4. Different levels of users: Administrator or regular user
5. Create account and flash card sets

Project objectives:

1. Create an interface for account creating
2. Extend studying techniques available through self-quizzing
3. Create random pick of what flashcard appears when studying.
4. Allow user to share their cards with people and user can apply owner, copy, and use privileges.
5. Changes made on copied set do not affect the original set.
6. Make web version that appears on mobile appear well on mobile device using a responsive design.

1.3 Project Deliverables

1. Accessible website with the ability to make an account.
2. Ability to make flash cards set.
3. Ability to edit/add/delete cards in a set.
4. Create sharing ability for flashcard sets.
5. Create a user guide.
6. Create separate administrative and user privileges.
7. Create a test plan for the application.

1.4 Assumptions and Constraints

Assumptions:

1. Format of the project is in HTML
2. Database is written in SQL
3. Users will know the basics of how to create an account and make a set.
4. The development team must meet the project requirements laid out in the project assignment.
5. Development location will change and developers will not be in the same place while developing which could slow down development time.

Constraints:

1. The software must run on a compatible PC, laptop, or mobile phone.
2. The database must be open source.
3. The software must be ready by 5/6/2019.
4. Approval is required from UMKC to make this an official UMKC app. Any UMKC symbols used in the app must adhere to guidelines the university provides.

1.5 Schedule and Budget Summary

02/18/2019— Iteration 1 Begins
02/22/2019 - Project Charter
03/01/2019— Requirements Document Baselined
03/04/2019— Iteration 2 Begins 03/04/2019— Iteration 1 Closeout
03/04/2019 - Project Plan
03/18/2019— Technical Prototype
03/19/2019— Iteration 3 begins
03/19/2019— Iteration 2 Closeout
03/25/2019— Project Status
04/05/2019— Architecture Document
04/09/2019— Iteration 4 Begins
04/09/2019— Iteration 3 Closeout
04/15/2019— Iteration 5 Begins
04/15/2019— Iteration 4 Closeout
05/03/2019 - Test Plan
05/05/2019— User and System Guide
05/06/2019— Iteration 5 Closeout
05/06/2019 Project Release

1 project manager at 4 hours per week for 14 weeks:
56 hours * $50/hr-$2800
1 requirement engineer at 4 hours per week for 14 weeks:
56 hours * $40/hr — $2240
2 software engineers at 4 hours per week each for 14 weeks:
112 hours * $40/hr $4480
224 hours total, $9520 total, avg, $42.50 per hour

1.6 Success Criteria

- The team delivers an operational prototype at the end of the semester with the features mentioned in the goal section above
- 80% or more of the team members would be willing to work together on another software project in the future.

1.7 Definitions

Agile Methodology: Agile methodology is a type of project management process, mainly used for software development, where demands and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customers.

1.8 Evolution of the Project Plan

At the beginning of each iteration the team will evaluate their goals from the last iteration and how well we were able to complete these goals. If we were able to complete the goals easily we will adjust the amount of work we take on in the next iteration and vice versa if the goals for the iteration were not met.

Risk mitigation efforts will be evaluated at the start of each iteration. Severe risks will be analyzed and added to the project plan as soon as they materialize.

2 Startup Plan

2.1 Team Organization

Project Managers (2):	The project managers are responsible for creating the project plan (with input from those doing the work), managing risks, running the weekly team meeting and providing monthly status reports to senior management.
Programmers (2):	Programmers are primary responsible for coding and unit testing modules. They are also expected to take part in architecture planning and review meetings.

2.2 Project Communications

There are three ways of communication:

1. Slack, online website used to communicate to group members as well as let advisor have access to conversation and any file transferring
2. Trello, used for displaying tasks to do, are currenting be worked on, and what is completed.
3. Github, repositories are pushed onto a single capstone project and allow for contributions to be seen.

2.3 Technical Process

This section describes the software development methodology or conventions the team agrees to live by. When following an organization standard process, this section will refer to the standard process and state any deviations that are planned for this project. In the absence of an organization standard process, this section will define planned phases, entry and exit criteria for each phase, major milestones, workflows, and other aspects of the proposed development process.

2.4 Tools

- Programming Language – C#
- Database – SQL
- Version Control – source code and written artifacts will be stored in a Subversion repository.
- Visual Studio
- Visual Studio Code
- Github
- Slack
- Trello

3 Work Plan

3.1 Activities and Tasks

A work breakdown structure is an excellent tool for identifying a complete list of tasks.

Depending on the needs of the project, some or all of the following attributes will be recorded for each task:

- Task name
- Task Description
- Owner
- Effort estimate
- Actual effort
- Planned start and stop dates
- Actual start and stop dates
- Dependencies among other tasks

3.2 Release Plan

3/18/2019— Technical Prototype
5/03/2019— Test Plan
5/05/2019— User and System Guide

3.3 Iteration Plans

2/18/2019— Iteration 1 Begins
3/04/2019— Iteration 2 Begins
3/04/2019— Iteration 1 Closeout
3/19/2019— Iteration 3 begins
3/19/2019— Iteration 2 Closeout
4/09/2019— Iteration 4 Begins
4/09/2019— Iteration 3 Closeout
4/15/2019— Iteration 5 Begins
4/15/2019— Iteration 4 Closeout
5/06/2019— Iteration 5 Closeout

3.4 Budget

1 project manager at 4 hours per week for 14 weeks:
56 hours * $50/hr-$2800
1 requirement engineer at 4 hours per week for 14 weeks:
56 hours * $40/hr — $2240
2 software engineers at 4 hours per week each for 14 weeks:
112 hours * $40/hr $4480
224 hours total, $9520 total, avg, $42.50 per hour

4 Control Plan

4.1 Monitoring and Control

Weekly	–	Team meeting. Project participants report status, progress and potential problems.
3/25/2019	–	Project status
4/5/2019	–	Architecture Document
5/3/2019	–	Test Plan
5/6/2019	–	Project release

4.2 Project Measurements

Phase	Measurement	Source
Release Planning	Record effort estimates for product features	Mgr
Iteration Planning	Record effort estimates for scheduled tasks Update effort estimates for product features Update estimated dates in release plan	Mgr
Iteration Closeout	Record actual effort for scheduled tasks Record actual effort for product features Record LOC count for modules written	Mgr/Pgr
System Test	Record the rate at which errors are found.	Pgr
Project Closeout	Archive project performance data in process database. (See process database definition for a list of measures to record.)	Mgr
Ongoing	Record defects found from integration testing through first year of release. Assign each defect to one of the following categories: blocker, critical, major, minor or trivial. Keep track of the state of each defect: open, assigned, fixed, closed.	Mgr/Pgr

5 Supporting Process Plans

5.1 Risk Management Plan

1. Establish likely risks in the project. Sit down as a group and discuss issues that may occur throughout the project and how as a team we can minimize these risks.
2. Once potential risks have been identified assess the risks. How likely is it to occur, what damage will it do, can we lower the likelihood through certain practices?
3. Learn how to manage the risk. Asa team discuss the risks we will take on in this project and how we should manage each one. Discuss what technique should be used for each risk: Avoidance, reducation, retention, and transfer.
4. Create a risk management plan. Select appropriate measures to minimize each of the risks.
5. Implement your risk management strategy. Take all the above steps and put them into practice.
6. Evaluate and review. Once these practices have been implemented come back to this process after some time and determine whether new steps can be implemented or whether certain risk management techniques need to be reevaluated.

5.2 Configuration Management Plan

1. All work files will be stored in our github page and updated documents and code will be posted there.
2. We will use version control in Github by committing code changes to a master branch. All changes to the code will be made in branches of the master branch and once they have been tested and approved these changes will be committed to the master branch.
3. The master branch will be the baseline for the source code. All other branches can be changed and must be reviewed before being committed.
4. Steps to committing to master branch:
   a. Create your own branch of the master branch.
   b. Make the code changes needed.
   c. Test code changes by running unit tests and manual unit tests.
   d. Have the code reviewed by another developer.
   e. Document code changes and make code comments in order to make other users aware of what changes were made.
   f. Commit code to master branch.
5. Change history will also be required for documents related to the project. Whenever a document is changed users will make changes and create and saved copy of the original document before their changes were made. This helps to allow version control on other documents related to the project.

5.3 Verification and Validation Plan

To verify to acceptability of the project we will perform unit testing and manual unit testing in order to determine if the product fulfills the requirements listed below.

5.4 Product Acceptance Plan

An acceptable product for launch is one that fulfills all the requirements listed in the project assignment.

1. Allow users to create an account and create flash card sets.
2. Creator can edit/add/delete from a set of flashcards and share the flashcards.
3. When a user studies a set the card is randomly selected and allows the user to guess the other side before flipping the card.
4. Stats of % are kept in history to show user improvement.
5. User can share flash card sets and apply owner, copy, and use privileges.
6. Copy rights allow people to copy card sets as their own.
7. The changes in copied sets will not affect the original set.