California State University Long Beach

College of Engineering

Computer Engineering Computer Science Department

Tuesday, November 12th, 2019

Sixfold Development Project Plan V2.0

Group Members:

Michell Kuang (Team Lead)	**013421094**
Joshua McDaniel	**014542786**
Jingyan Du	**014436615**
Peter Park	**002948398**
Jacen Tan	**012393782**
Daniel Gione	**016513144**

Version History

Version	Date	Author(s)	Comments
V1.0	10/10/19	All members	-Original document
V2.0	10/18/2019	Jingyan Du	-Document revise -Risk Management updated -Resource updated -Added a TimeLine graph -Schedule, sprint plan, milestones updated

Table of Contents

Overview

This document aims to establish a working timeline to keep developers on track with creation of features. It explores the risks undertaken within the project, such as scope/requirements changes, developer injury and software changes. It also helps to establish a scope and calculates cost estimates of the project as a whole, separating items such as labor, software, and testing.

Project Goal

Goal	Priority	Description
Assist with managing household responsibilities	High	Solve the problem of responsibilities management on cohabiting people
Assist with matching people looking to cohabitate	Low	Help household recruitment and searching household

Project Scope

- **User Scope**
  - Our target users are adults who need to share a house or apartment with other people.
  - The regional restriction of our product is the whole California.
- **Application type**
  - Our product will be a single page web application
  - Users can access our application by Internet connection (broadband) with minimum speed of 30mbps:
    - Wired Internet - Allow users to access the application by cabled Internet.
    - Wireless Internet - Allow users to access the application by WIFI.

Evolution

- As one the necessary design documents, the project plan should always been updated during the development of the project.

Resources

- Developers - Six undergraduate students from California State University Long Beach

    - Michell Kuang - Team Lead
    - Joshua McDaniel
    - Jingyan Du
    - Peter Park
    - Jacen Tan
    - Daniel Gione

  All team members are developers and testers.
- Time - Estimate time: 1159 hours for the entire project. The estimate time will change due to any changes of scope/requirements/plans/risk.
    - Implementation start from 10/18/19 to 5/7/2020 - Total 203 days
- Funding - Estimated: $150000
    - Equipment - Developers are using their own laptops for project.
    - Print - Since most documents are required to be printed out. We make an estimated cost for printing.
    - Labor
        - Developers - we assume we will be working 6 hours a day, for 215 days, at 18 dollars an hour for 6 people it ends up becoming about 6 X 215 X 18 X 6 = 139320 dollars.
        - System administrator for managing databases and different accounts - 18 dollars an hour, 6 hours a day, 5 days a week indefinitely - $11286
    - Software Cost

Technology Type	**Software**	**Version**	**Cost**
IDE	Visual Studio Community	V16.3.29418.71	$0
Source Code Editor	Visual Studio Code	V1.39.2	$0
Team Communication	Discord	V0.0.305	$0
Mail	Gmail	NA	$0
Project Documentation	Google Drive	NA	$0
	Draw.io	NA	$0
	Time.graphics	NA	$0
Project Management	GitHub	NA	$0
Web Development	Google Chrome	V63.0.3239	$0

Web Server	Microsoft Internet Information Services (IIS)	V10.0.18362.1	$0
Cloud Computing Service (Web Server Host)	Google Cloud Platform	NA	$0($500 coupon per month, shared with the whole class)
Web Frameworks	.NET Framework	V4.7.2	$0
	ASP.NET	V4.7.1	$0
	Vue.js	V2.6.10	$0
Relational Database	Structured Query Language (SQL) Server	2017 Developer Edition	$0
	Structured Query Language (SQL) Server Management Studio (SSMS)	V. 18.3.1	$0

Most software we are using are free but an estimated cost for software as $100 is required for software or technology changes.

Total Cost

Cost	Expense
Labor	$139320+$11286 = 150606
Software Cost	$100
Equipment	$0
Print	$200
Total	$150900

Risk Management

This will be a long-term and stressful project, so in order to ensure that the project is completed on time, we must be prepared to deal with possible risks.

Whenever a risk is issued:

- Evaluate the risk level.
- Try to complete current sprint.
- Update all documents with effects by the risk.
- Calculate the new estimate project due date with effects by the risk.
- Deal with the risk.

Solution to Risks:

- Weekly meeting - let the whole team know if a risk is required to be handled.
- Unallocated Time - We leave 9 unallocated sprints before the project due date for buffer, started after sprint 5.
- Communication with the client - always communicate with the client when it is necessary.
- Flexible developers - tasks shall be changed depends the living project plan.
- Living documents - documents will always be updated
- Alternate software -
- Backup - code for implementations and documents shall be backuped after each sprint is finished.

Possible Risks:

- **Scope / Requirement Changes** - when the client requests a change on scopes or requirements:
    - Provide the client a more efficient alternate plan if possible.
    - **Low Risk Level** - The changes of scopes/requirements caused no significant effects on the plan.
        - Start working for the next sprint with the new plan.
    - **Medium Risk Level** - The changes of scopes/requirements required at least one extra sprint to finish the project.
        - Modify the plans and use unallocated time for new tasks.
    - **High Risk Level** - The changes of scopes/requirements required more than 6 extra sprints to finish the project.
        - At least 3 unallocated sprints should be available for other possible risk, the developers must notify the client that the changes may bring more risks in the future. And ask if the client want an alternate plan.
    - **Buffer Limit** - The changes of scopes/requirements required more than 9 extra sprints to finish the project.
        - Notify the client that the changes shall cause the project unable to be complete as planned.
        - Offer the client two options:
            - Delay the project due date.
            - Alternate plan for requirements/scope changes or cancel them.

- **Developer Illness/Injury** - when developers cannot work because of uncontrollable factors:
    - Weekly Meetings
        - Check if the team members can continue to work normally every week.
    - Split Tasks
        - Start a group meeting and change project plan based on affected tasks.
        - Assign tasks that require 10 hours to other available members.
    - **Low Risk Level** - The developer can still work on part of his tasks.
        - Switch easier tasks to certain developer.
        - Split remained tasks to other developers.
    - **Medium Risk Level** - The developer is totally unable to work.
        - Split tasks to other developers.
        - Modify the plans for remaining tasks.
        - Remaining tasks shall have the highest priority for the next sprint.
        - Notify the client that current sprint is unable to be completed as planned due to developer illness/injury.
    - **High Risk Level** - Two developers are totally unable to work.
        - Modify the plan, use unallocated time and assign remaining tasks to the next sprint.
        - Notify the client that current sprint is unable to be completed as planned due to developer illness/injury.
    - **Buffer Limit** - When more than 2 developers are totally unable to work.
        - Notify the client the project unable to be complete as planned.
        - Offer the client two options:
            - Delay the project due date.
            - Alternate plan to decrease time cost.

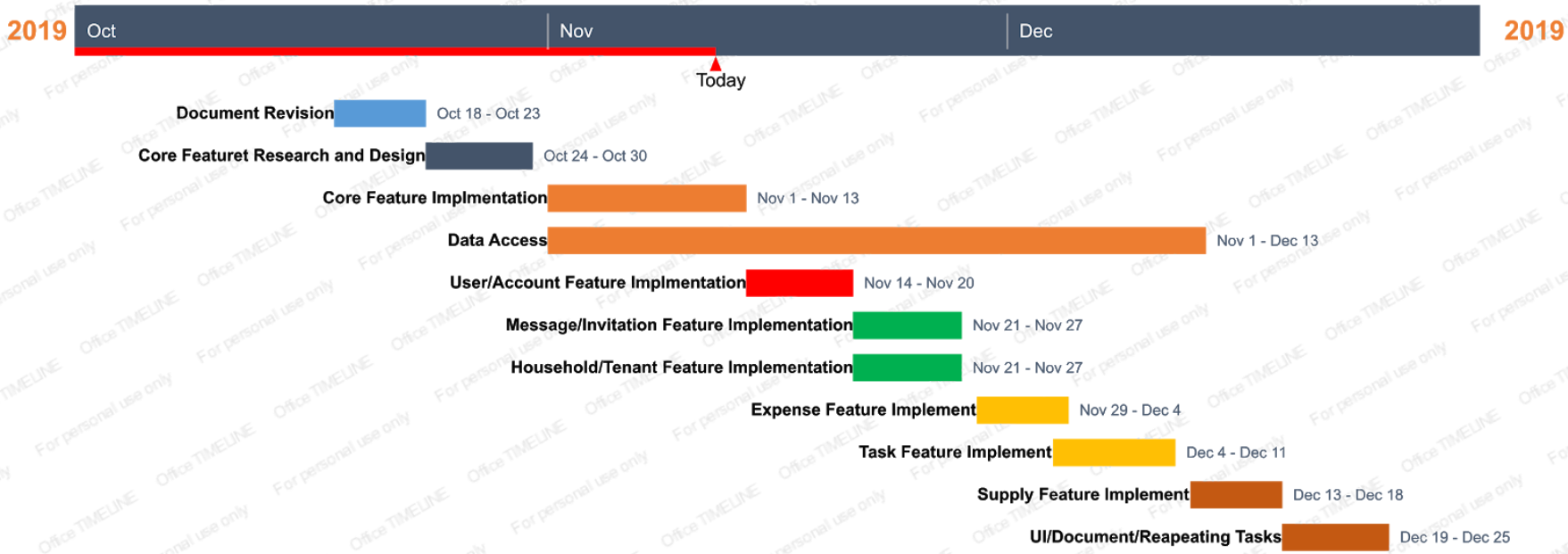
- **Software Changes -** whenever current libraries or frameworks cannot satisfy the project's requirement or a new version stopped supporting our features:
        - **Low Risk Level** - The developer can still work on the software after downgrade it to original version or the alternate software won't cost significant extra time.
            - Continue work as planned, modify the plan based on new software if needed.
        - **Medium Risk Level** - The developer can work no more on the original software, but the alternate software won't cost significant extra time.
            - Modify the plan based on new software.
            - Continue work on the original software or alternate software if needed.
        - **High Risk Level** - The developer can work no more on the original software, and the alternate software cost significant extra time.
            - Modify the plan based on new software, use unallocated time for extra time cost.
            - Move the project to the alternate software.
            - Notify the client if the software changes shall cause a failure for current failure.
        - **Buffer Limit** - when using an alternate software shall cause project incomplete.

            - Notify the client the project unable to be complete as planned.
            - Offer the client two options:
                - Delay the project due date.
                - Alternate plan to decrease time cost.
- **Cost Changes** - whenever a free software is not free anymore.
    - Free developing tools to save on budget - most software we are using for this project are free which shall not be affected by the cost changes.
        - **Buffer Limit** - when the software cost more than $100
            - Since the budget for software cost is $100, when the software requires more than $100, we shall classify it as **Software Changes** risk

Project Schedule

- The project shall be managed using scrum, we shall use sprints to divide the tasks of the entire project. The plan for sprints is estimated and will be updated as the project is developed.
- **Unallocated Time** - We leave enough time for risk management. After sprint 5, unallocated time will be used for:
    - Plan changes
    - Scope/Requirement changes
    - Developer injury/illness
    - Software changes
    - Cost changes
- The estimated timeline:

TimeLine


2019
Oct
Nov
Dec
2019
Today
Document Revision
Oct 18 - Oct 23
Core Featuret Research and Design
Oct 24 - Oct 30
Core Feature Implmentation
Nov 1 - Nov 13
Data Access
Nov 1 - Dec 13
User/Account Feature Implmentation
Nov 14 - Nov 20
Message/Invitation Feature Implementation
Nov 21 - Nov 27
Household/Tenant Feature Implementation
Nov 21 - Nov 27
Expense Feature Implement
Nov 29 - Dec 4
Task Feature Implement
Dec 4 - Dec 11
Supply Feature Implement
Dec 13 - Dec 18
UI/Document/Reapeating Tasks
Dec 19 - Dec 25

Sprint Plans

- After the end of a sprint, one day will be reserved for sprint review analysis and client demo. For the sprint review analysis, a group meeting is required to assign tasks, evaluate exact velocity and update the plan based on analysis.
- The velocity of the developer team should keep increase as developers are being better working with scrum.
- New sprints will be discussed and updated when add-on tasks issued or the plans were changed.

Sprint NO.	**Start Date**	**End Date**	**Planned Velocity**	
1	10/18/2019	10/30/2019	203	
Deliverable	**Priority**	**Complexity**	**Effort**	**Hours**
Revise BRD (V.2)	10	Medium(3)	13	26
Revise Tech Spec (V.2)	11	Medium(3)	13	26
Revise Design Document (V.2)	12	Easy(2)	5	10
Revise Project Plan (V.2)	13	Easy(2)	5	10
Revise Project Roadmap (V.2)	14	Easy(1)	0.5	1
Revise Sitemaps (V.2)	15	Easy(1)	2	4
Error Handling Research and Design	16	Medium(3)	13	26
Logging Research and Design	17	Easy(2)	8	16
Data Store Archiving Research and Design	18	Medium(4)	13	26
Flat File Archiving Research and Design	19	Medium(3)	8	16
AuthN Research and Design	20	Medium(3)	13	26
AuthZ Research and Design	21	Medium(3)	8	16

Sprint 1

- Revise all documents - good documents are always necessary for a developer team. Since our project is based on there documents, we must have good enough documents before we actually start working on implementation.
- Core Requirements - features we need to finish for milestone 2
    - Error Handling Research and Design
    - Logging Research and Design
    - Data Store Archiving Research and Design
    - Flat File Archiving Research and Design
    - AuthN Research and Design
    - AuthZ Research and Design
- Since this sprint will be the first formal sprint, we shall use experiences from this sprint to help future plan. So, the velocity for this sprint won't be fully used.

Sprint NO.	**Start Date**	**End Date**	**Planned Velocity**	
2	11/1/2019	11/13/2019	242	
Deliverable	**Priority**	**Complexity**	**Effort**	**Hours**
Error Handling Implementation and Testing	22	Medium (3)	13	26
Logging Implementation and Testing	23	Easy (2)	8	16
Data Store Archiving Implementation and Testing	24	Medium (4)	21	42
Flat File Archiving Implementation and Testing	25	Medium (3)	8	16
AuthN Implementation and Testing	26	Medium (4)	21	42
AuthZ Implementation and Testing	27	Medium (3)	13	26
Data Store Access	28	Hard (5)	34	68

Sprint 2

- Core Requirements - implementation and testing
    - Error Handling Research and Design
    - Logging Research and Design
    - Data Store Archiving Research and Design
    - Flat File Archiving Research and Design
    - AuthN Research and Design
    - AuthZ Research and Design

  Necessary for milestone 2 and the entire project, finish the implementation and testing for these features shall always be the most important.
- Data Store Access
    - Data Storage is necessary before implementing any product feature.

Sprint NO.	**Start Date**	**End Date**	**Planned Velocity**	
3	11/15/2019	11/27/2019	248	
Deliverable	**Priority**	**Complexity**	**Effort**	**Hours**
System Admin	29	Easy (1)	3	6
Create Account	30	Easy (1)	5	10
Delete Account	31	Easy (1)	5	10
Enable/Disable Account	32	Easy (1)	8	16
Registration (normal Users)	33	Medium (3)	13	26
User Profile	34	Easy (1)	8	16
Messaging Feature	35	Hard (5)	13	26
Household Creation	36	Medium(3)	8	16
Household Profile	37	Easy (1)	8	16
Household Search	38	Hard (5)	21	42
Invitation	39	Medium(3)	13	26
List specific tenants	41	Medium(4)	13	10
Tenant Promoting	42	Medium(3)	3	6
Tenant Demoting	43	Medium(3)	3	6
Assign Host	44	Medium(4)	8	16

Sprint 3

- User Account
    - System Admin
    - Create Account
    - Delete Account
    - Enable/Disable Account
    - Registration (normal Users)
    - User Profile
- Household system - Household is necessary since all managements are based on a household.
    - Creation
    - Profile
    - Search
- Tenant Management - Tenant management is required for assigning tasks, split expenses and upload supply request.
    - List
    - Promotion
    - Demotion
    - Assign Host
    - Household disbandment * to fit the team velocity better, this feature was assigned to the next sprint.
- Messaging Feature - Messaging system helps users communicate and schedule interviews with host of a household.
- Invitation Feature - A user shall be able to join a household by invitation to fully access features of this program.
  Necessary features for product. Users shall only access any features with an account.

Sprint NO.	**Start Date**	**End Date**	**Planned Velocity**	
4	11/29/2019	12/11/2019	256	
Deliverable	**Priority**	**Complexity**	**Effort**	**Hours**
Household disbandment	45	Hard(5)	13	26
ExpenseCreation	46	Medium(3)	8	16
Expense Viewing	47	Easy2	3	10
Expense Editing	48	Medium(3)	8	16
Expense Deletion	49	Easy(2)	3	10

Expense Billing Cycle	50	Hard (5)	13	26
Expense Splitting	51	Hard (5)	13	42
Task Creation	52	Medium(3)	13	16
Task Viewing	53	Easy (2)	3	10
Responsible Tenant List	54	Medium(3)	13	16
Task Editing	55	Medium(3)	13	16
Task Deletion	56	Easy (2)	8	10
Task Notification	57	Medium(3)	13	26
Task Completion	58	Medium(3)	13	16

Sprint 4

- Household disbandment - A host shall be able to disband a household if he/she is the only tenant in this household.
- Expense Management System:
    - Creation
    - Viewing
    - Editing
    - Deletion
    - Billing Cycle
    - Splits

- Task Management System:
    - Creation
    - Viewing
    - Responsible Tenant List
    - Editing
    - Deletion
    - Notification
    - Completion

Sprint NO.	**Start Date**	**End Date**	**Planned Velocity**	
5	12/13/2019	12/25/2019	210(Shall be updated based on changes of the plan)	
Deliverable	**Priority**	**Complexity**	**Effort**	**Hours**
Repeating Tasks	59	Medium(3)	13	26
Privacy	60	Hard (5)	21	42
Supply Request Creation	61	Medium(3)	13	16

Supply Request Viewing	62	Easy(2)	3	10
Supply Request Editing	63	Medium(3)	13	16
Supply Request Deletion	64	Easy (2)	8	10
UI/UX	65	Medium(3)	21	42
User Manual	66	Easy(1)	13	16
FAQ	67	Easy(1)	13	16
Developer Docs	68	Easy(1)	13	16

Sprint 5

- **Unallocated time** shall be used for any plan changes and risk management.
- Privacy - Data safety and restricting the amount of information available to the public.
- Supply Request Management System:
    - Creation
    - Viewing
    - Editing
    - Deletion
- User Manual - The important document to help user understand and use this program.
- FAQ - The document lists all common problems related to this program, problems shall be collected from unit tests.
- Developer Docs - The important document to help developers understand this program functionally for better updating and maintenance.

MileStones

Milestone 1

Date: 11/28/2019
Objective:

- Complete implementations for:
    - Core Requirements
    - Data Store Access
    - User Account
    - Household System
    - Task Management System
    - Message Feature
    - Invitation

- Modify the plans based on any changes of scope/requirements/expected risk.
- Documents updated

Milestone 2

Date: 12/26/2019
Objective:

- Complete implementations for:
    - Household disbandment feature complete
    - Expense Management System
    - Task Management System
    - Privacy
    - Supply Request System
    - UI/UX
    - User Guide Documents:
        - User Manual
        - Developer Docs
        - FAQ
- Modify the plans based on any changes of scope/requirements/expected risk.
- Documents updated

Project Due Date

Date: 5/7/2020
Objective:

- The project must be complete
- All features work functionally
- All tests for the entire project are finished