Software Project Plan

Nick Azzouz, Christian Balan, Leonardo Diaz, Xavier Evans, Zane Richards

Table of Contents

1. Introduction

The SplitSmart application will be a software that can be used to track and share expenses between users. This allows for simple and easy deliberation between groups when tracking and making payments. SplitSmart can be accessed via web browser, or the mobile app allowing users to be on top of their expenses wherever, whenever. This document will provide an in depth overview of the tasks, timeline, risks, and staff that will be involved in the creation of the SplitSmart application.

1.1 Project Scope

The following features will be included in the final version of the application:

Group **creator** should be able to:

- Create an account and sign in.
- Add/remove members to/from a group or contact list.
- Create transactions within the group and with contacts.
- Choose the desired payment split-method per transaction.
- Cancel payment request.
- Label transactions with a name (e.g. Pizza Party).
- Provide transactions with relevant images.
- Review timestamped transaction history and receipts.
- Receive email of transaction receipts, including remaining balance.
- Record and acknowledge a cash transaction from a member toward an unsettled balance.

Group **members** should be able to:

- Create an account and sign in.
- Review timestamped transaction history and receipts.
- Receive email of transaction receipts, including remaining balance.
- Comment (acknowledge) an outstanding transaction.
- View settled and unsettled expenses.
- Receive email of an owed/satisfied balance.
- Receive a notification of a failed notification.
- Pay a transaction with paypal, venmo.
- Record a cash transaction toward an unsettled balance.

1.2 Major Software Functions

- **Account Registration:** The account registration function will allow users to create accounts. To set up an account the users should provide their full name, home address & email, and desired payment information.
    - Name (first, last)
    - Mailing address
    - Building address
    - Email address
    - Confirm email address
    - Credit card info or online wallet (maybe option to use checking account)
        - Give an alternative option to ‘Record a Cash Payment’
    - Password
- **Update Account Information:** The Update Account Information will give users access to edit their information. They can modify name, mailing/billing address, email address, payment methods, and password.
    - Name (first, last)
    - Mailing address
    - Building address
    - Email address
    - Payment Method(s)
    - Password
- **Account Login:** The Account Login function will allow users to access their account via a username and password. Once a user’s login information has been verified, they will be able to make changes to their account such as changing their password.
    - User ID
    - Password
- **Account Logout:** The Account Logout function will allow the user to exit their account.
- **Expense Creation:** The Expense Creation function will allow users to create an expense for either themselves or as a group.
    - Parties involved
    - Payment info
    - Date
    - Expense amount
    - Payment split
    - Receipt
    - Balance change
    - Expense approval

- **Create Group:** The Create Group function will allow users to create groups to allow users to track shared expenses and balances easier
    - User IDs
    - Expenses
    - Balances
- **Edit Group:** The Edit Group function will allow users to edit the users that are in the group. It will allow members removed to be involved in expenses that are still ongoing, but nothing further.
    - Parties Involved
    - Date
    - Expense

- **Delete Group:** If all balances are settled, the delete group function will allow users to delete an existing group and make the transaction history for that group available via emailed PDF upon request.
    - Group ID
- **Invite Friend:** The Invite Friend function will allow the user to enter in the person's email that they wish to be friends with on SplitSmart. Then an invite request will be sent to the recipient's email address.
    - Going to
    - Coming from
- **Remove Friend:** The remove friend function will allow the user to remove friends from their friends lists and only their friends list. (user remains in existing groups).
    - User ID
- **Notification:** The Notification function will send the user an alert when an expense is made that they are involved in.
    - Expense name
    - Group ID
- **Balance tracking:** The Balance Tracking function will keep track of payments made by users within a group and display the most up-to-update balances for each user next to their account.
    - User ID
    - Balance
    - Expenses
- **Payment Tracking:** The Payment Tracking function will allow the user to track expenses linked with the user's account. It will update the User's balance when they pay an expense or are paid for an expense.
    - User ID
    - Group User IDs
    - Payment Info

    - Expenses
    - Balance
- **Report:** The Report function will allow the user to see a summary of balance changes, payments made, and expenses they were involved in along with the dates where the changes occurred or other filters.
    - User ID
    - Expense
    - Balance
    - Payment
    - Date
    - Group
    - Filter Option

1.3 Performance/ Behavior Issues

The biggest concern when looking at performance is high traffic especially with many users within a group. Depending on the time of year the app may experience various amounts of traffic, for example during summer when vacations are more common there may be a higher amount of groups being created and managed. It is crucial, especially when dealing with finance that the application functions correctly and without issues. It is also important that both the mobile, and web versions of the application maintain a consistent user-friendly interface that can be easily navigated between devices.

1.4 Management and Technical Constraints

The client has requested that SplitSmart should allow users to create and manage groups of people. The functionality of allowing groups in the app as opposed to only 2 people splitting expenses will require the software team to need more time. The client hasn't proposed a deadline yet but we are using our time wisely. The software team understands time is a limited source and the team will strive to stay on schedule and update the client as much as possible. The team consists of only 5 people so developing a big scale project will be difficult to complete if time given to us is short. Additionally, since the developers are relatively inexperienced, there must be time dedicated to research the various tools that are going to be used, thus taking up more of our time.

2. Project Estimates

2.1 Historical data used for estimates

Using data found from Salaray.com for the job titled “Web Developer” in the state of Michigan we have the following pay range:

Low = $60,149
Median ≈ $70,000
High = $80,377

Estimated Labor rate per person-month:
($70,000/12) **= $5,833**

2.2 Estimation techniques applied and results

2.2.1 FPA Based Estimation:

External Inputs (EI)
1. Account Registration
2. Update Account Information

External Outputs (EO)
1. Balance Tracking
2. Payment Tracking
3. Notification

External Inquires (EQ)
1. Account Login
2. Account Logout
3. Expense Creation
4. Create Group
5. Edit Group
6. Delete Group
7. Invite Friend
8. Remove Friend

Internal Logical Files (ILF)
1. Account Information
2. Report
3. Database

External Interface Files (EIF)
1. N/A

Function Points

Information Domain Value	Count	Weighting Factor	Total
External Inputs (EI)	2	4	8
External Outputs (EO)	3	7	21
External Inquiries (EQ)	8	8	64
Internal Logical Files (ILF)	3	5	15
External Interface Files (EIF)	0	0	0
Total			108

FP Value Adjustment Factors (0 [not needed] - 5 [absolutely necessary])

1. Backup required ? 5
2. Data communication ? 5
3. Distributed processes ? 5
4. Optimal performance ? 3
5. Heavily used operating systems. ? 3
6. On-line data security ? 5
7. Multiple screens ? 3
8. On-line master file update ? 3
9. Complex inputs, queries, outputs ? 3
10. Complex internal processing? 3
11. Reusable code ? 0
12. Conversion or installation in design ? 0
13. Multiple user sites ? 0
14. Ease of use ? 5

Total = 43

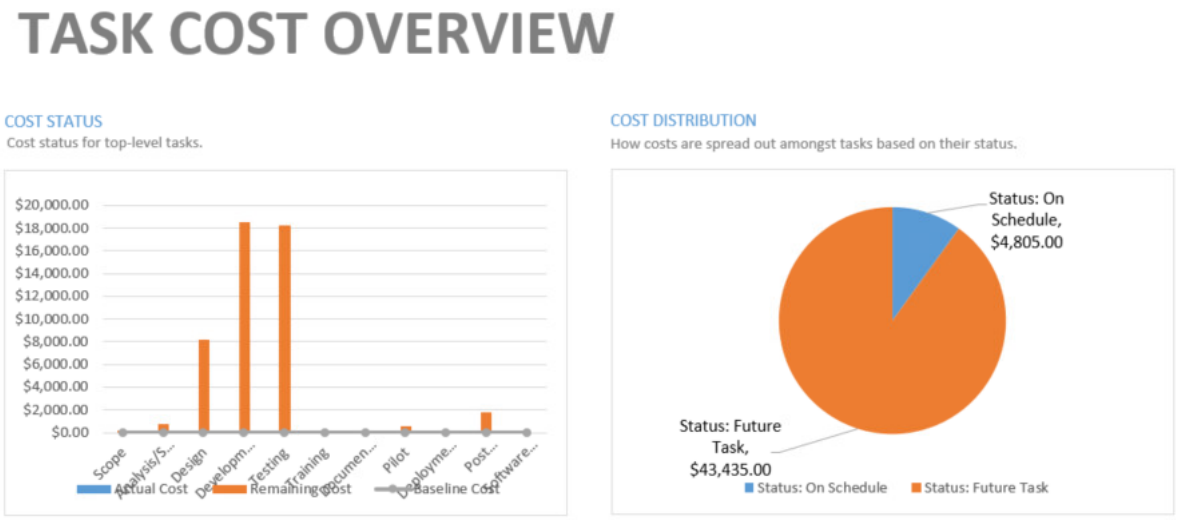
Value Adjustment Factor

$[0.65 + 0.01 * S(F_i)] = [0.65 + 0.01 * (43)] = 1.08$

$FP = T * (0.65 + 0.01 * \sum F_i) = 108 * 1.08 = 116.64 = 117$

Average Productivity = 15 FP/pm
Burdened Labor Rate = $5,833 per month
Cost per FP = ($5833 / 15) = $388 per FP

2.2.2 Process Based Estimation:



COST DETAILS

Cost details for all top-level tasks.

Name	Fixed Cost	Actual Cost	Remaining Cost	Cost	Baseline Cost	Cost Variance
Scope	$0.00	$0.00	$235.00	$235.00	$0.00	$235.00
Analysis/Software Requirements	$0.00	$0.00	$725.00	$725.00	$0.00	$725.00
Design	$0.00	$0.00	$8,200.00	$8,200.00	$0.00	$8,200.00
Development	$0.00	$0.00	$18,480.00	$18,480.00	$0.00	$18,480.00
Testing	$0.00	$0.00	$18,200.00	$18,200.00	$0.00	$18,200.00
Training	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00
Documentation	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00
Pilot	$0.00	$0.00	$600.00	$600.00	$0.00	$600.00
Deployment	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00
Post Implementation Review	$0.00	$0.00	$1,800.00	$1,800.00	$0.00	$1,800.00
Software development template complete	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00

2.3 Reconciled Estimate

Final Estimated Project Cost: (cost per FP x FP) = $45,396
Final Estimated Effort: (FP / average productivity) = 117 / 15 = 7.8 person-months
Final Estimated Duration: (person months / persons) = 7.8 / 5 = 1.56 months

2.4 Project Resources

People:

- 5 CECS students at the University of Michigan Dearborn
- Members have experience with C++ and Python programming languages
- Front-end web development is new to most members

Hardware:

- Personal computers

Software:

- Google suite applications
    - Sides, Docs, Calendar, etc
- Zoom

    - Used for weekly meetings
- Enterprise Architect
    - Used for UML Diagrams
- Microsoft Project
    - Use for project timeline estimations

3. Risk Management

3.1 Project Risks

We divide the following risks into various risk identification categories:

1. Risks of **Project Size**
    a. Given that the amount of transactions stored by groups and its members are highly variable, there is a concern that the system will not be scalable enough to accommodate a growing number of stored transactions.
        i. Transaction storage period may be dependent on members' tax requirements.
2. Risks of **User Experience**
    a. For less technically literate users, there is concern that the app may be too difficult to use, resulting in revisions that may be costly.
    b. Given that the app will be developed for web use, there is concern about how extensible it should be to run on most major devices (smartphones, personal computers, etc.)
3. Risks of **Developers:**
    a. Given that this software requires developers to put forth effort towards its completion, there is a concern that developers will lose motivation and as a result lose productivity.
    b. Given that no member of the software development team is an expert in web development, there is a concern that lack of experience could cause delays in the creation of the software.
    c. Given that the software development team is small, there is a concern that given the short timeframe not all aspects of the software can be completed in time.
    d. Given that the software development team consists of students, there is a concern that some members may have other class responsibilities which could cause slowdowns.
4. Risks of **Transaction Modification**

   a. In the event of a price adjustment of a posted transaction, there is a concern whether members who have already paid the creator of the transaction will receive the appropriate adjustment.
   b. A user may be concerned with the ability to record or accept cash transactions.
5. Risks of **Security Concerns**
   a. A user paying with their credit card or bank account may be concerned about the security of their transaction.
6. Risks of **Currency Conversion**
   a. Given that a payer's currency's value may change at any given moment will cause it to weigh differently against the payee's currency should they differ.
7. Risks related to **Payment Methods**
   a. Given that over time new online payment will rise and old ones may cease. There is concern regarding the cost and time required to implement new payment methods like (PayPal, Google, Apple, Venmo).

3.2 Risk Table

Risk Identification categories:

PS = **P**roject **S**ize Risk
UE = **U**ser **E**xperience Risk
TM = **T**ransaction **M**odification Risk
SC = **S**ecurity **C**oncern Risk
CC = **C**urrency **C**onversion Risk
PM = **P**ayment **M**ethod Risk
DR = **D**eveloper **R**isk

Risk Impact Index:

1 = Catastrophic
2 = Critical
3 = Marginal
4 = Negligible

Table is ordered by impact rating where **higher priority is at the top**.

The final column is a placeholder, pointing to the RMMM section that follows this section. We chose our scale for the risks to better focus on the most crucial risks that affect user interaction with the software.

Risks	Category	Probability	Impact	RMMM
Staff turnover	DR	10%	1	*
Lack of security of payment compromises user data	SC	15%	1	*
User payment method information are made accessible on their account (not hidden or encrypted)	SC	5%	1	*
Software bugs cause miscalculation during currency conversion	CC	20%	1	*
Most recent currency weights not pulled at time of transaction	CC	10%	1	*
Developing a GUI that meets the accessibility requirements of less technically literate users	UE	30%	2	*
User payment method is not supported	PM	15%	2	*
Web page is not fully supported by all major platforms and browsers	UE	17%	2	*
Team conflict	DR	5%	2	*
User paying through a service is denied payment due to product terms being out of date	PM	15%	2	*
Some team members are not available full time	DR	25%	3	*
Transaction database is not extensible enough to support future additions	PS	15%	3	*
Downtime caused by lack of storage hardware. (Unplanned scalability)	PS	20%	3	*
User complaints about lack of the latest visual styling standards in web interface	UE	10%	4	*

We selected “User paying through a service is denied payment due to product…” to be the cutoff risk. Risks following the aforementioned will not be actively monitored as risks below level 2 (critical) are less likely to occur.

3.3 Overview of Risk Mitigation, Monitoring, Management

Now that the risks of the project have been established, it is critical that the software development team takes a proactive approach to ensure that the previously mentioned risks do not threaten the completion of the project. This can be done by recognizing the causes of these risks and constantly monitoring for these causes throughout the lifespan of the project. The three main steps we will be using to achieve this goal is risk: mitigation, monitoring, and management.

Mitigation is the process of ensuring that we mitigate, or reduce the chances of any of the defined risks from ever occurring. Simply put, with mitigation, we aim to lower the probability of any risks occurring. Unfortunately it is not always possible to prevent risks and as a result, we need to have steps in place to manage these risks when they arise.

With Monitoring, the group will study the defined risks and assess the chances of them occurring. This also includes ensuring that the Mitigation strategies are doing what we intend them to do by reducing the chances of risks occurring. However, if during monitoring the group deems a risk has occurred, it is important to have a strategy to resolve that risk as quickly as possible.

Finally with Management we will look into what action can be taken in the event of a risk occurring. A game plan is created for each risk to ensure that the project can stay on task and on schedule and risks can be managed effectively and efficiently.

See below the Risk Information Sheets (RIS) which iterates through each risk and provides the Mitigation, Monitoring and Management for each of the defined risks.

Risk type: Developer Risk
Priority: 1
Risk factor: Staff turnover
Probability: 10%
Impact: Without full team support, SmartSplit will be impossible to complete

Mitigation Strategy: Ensure constant communication between team members and keep track of team member participation on various components of the project.
Monitoring Approach: To ensure stability, team morale must be evaluated on a regular basis.
Management Plan: Rapidly search for a replacement team member and contact the Professor.
Estimated resources: Extend project completion date to reconfigure teams and redistribute the workload with the fragmented team.

Risk type: Security Concern Risk
Priority: 1
Risk factor: Lack of security of payment compromises user data
Probability: 15%
Impact: Without payment security, users will not trust the website payment methods
Mitigation Strategy: Consult cybersecurity experts in the field to ensure the highest level of security can be achieved.
Monitoring Approach: Ensure payment security is a top priority when developing the software.
Management Plan: Apologize and refund any lost money to users if their credit card information was taken and rapidly develop new security.
Estimated resources: Since payment security is our top priority, the workload will be distributed to all 5 members of the team.

Risk type: Security Concern Risk
Priority: 1
Risk factor: User payment method information are made accessible on their account (not hidden or encrypted)
Probability: 5%
Impact: User's personal information will be exposed
Mitigation Strategy: Consult cybersecurity experts in the field to ensure the highest level of security can be achieved.
Monitoring Approach: Run tests to make sure user data is encrypted
Management Plan: Shut down the website to fix errors in the code or implement new code and run many tests.
Estimated resources: Workload will mainly focus on running tests to make sure user data is encrypted

Risk type: Currency Conversion Risk

Priority: 1
Risk factor: Software bugs cause miscalculation during currency conversion
Probability: 20%
Impact: Users can send or receive incorrect amounts of money as a result of a conversion bug.
Mitigation Strategy: Have banks deal with conversion rates or find a framework that is made with live, accurate conversion rates already in place.
Monitoring Approach: Run frequent test payments to ensure conversions work as intended.
Management Plan: Mandate the use of U.S currency only to bypass issues with currency conversion or leave conversions up to the banks so it doesn't have to be managed by the software.
Estimated resources: Time to run tests and ensure that there are no bugs causing calculation errors.

Risk type: Currency Conversion Risk

Priority: 1
Risk factor: Most recent currency weights not pulled at time of transaction
Probability: 10%
Impact: Users can lose money if most recent currency weights are not pulled at time of transaction
Mitigation Strategy: Create delay during time of currency weights being pulled to ensure transactions can occur at proper time.
Monitoring Approach: Partner with a bank when it comes to overseas transactions
Management Plan: If nothing works out, SmartSplit will only accept U.S currency
Estimated resources: A team member will be responsible for ensuring up to date currency weights

Risk type: User Experience Risk

Priority: 2
Risk factor: Developing a GUI that meets the accessibility requirements of less technically literate users
Probability: 30%
Impact: Our website will have bad reviews if less technically literate users cannot use our software
Mitigation Strategy: Utilize unfamiliar users to test the software during development to see what can be improved upon in the UI.

Monitoring Approach: Let family members who aren't tech savvy use our website for testing purposes
Management Plan: User feedback will be important for the team to know if the website is user friendly
Estimated resources: The group member in charge of design will be responsible for creating innovative unique ideas for the GUI

Risk type: Payment Method Risk
Priority: 2
Risk factor: User payment method is not supported
Probability: 15%
Impact: User is not able to complete payments through the app and may choose to stop utilizing the app entirely.
Mitigation Strategy: Only allow users who have a valid payment method to create an account.
Monitoring Approach: Ensure that the most commonly used payment methods such as Paypal, Google, Apple, Venmo, are all supported and all relevant services are not experiencing issues.
Management Plan: Ensure that it is clear to users what payment methods are supported when they are creating an account.
Estimated resources:

Risk type: User Experience Risk
Priority: 2
Risk factor: Web page is not fully supported by all major platforms and browsers
Probability: 17%
Impact: Considering almost the entire population on earth has a smartphone, we will lose out on users who strictly use smartphones, tablets, etc.
Mitigation strategy: Publish a mobile application of our website to IOS and Android.
Monitoring Approach: Our main concern would be to create a fully functional website before designing an application for mobile devices, and test on various devices regularly.
Management Plan: After our website is published, we can begin designing the mobile application
Estimated resources: Workload will be distributed to all 5 members to design a mobile application

Risk type: Developer Risk

Priority: 2
Risk factor: Team conflict
Probability: 5%
Impact: The project will be a total failure if team members don't get along
Mitigation Strategy: Communication
Monitoring Approach: Replace a team member if they don't want to communicate with the team.
Management Plan: Team members will need to speak up if they're upset with something and communicate with the team.
Estimated resources: Every member of the team is responsible for preventing any team conflicts

Risk type: Payment Method Risk
Priority: 2
Risk factor: User paying through a service is denied payment due to product terms being out of date
Probability: 15%
Impact: Users will be upset with SmartSplit and they will leave negative reviews
Mitigation Strategy: Create a pop up message that will let users know to check their service product terms before paying
Monitoring Approach: Apologize to user for any inconvenience
Management Plan: We will let users know that SmartSplit will not be responsible for any denied payment from a third party service
Estimated resources: A team member will be responsible for designing and implementing the message for users to know

Risk type: Developer Risk
Priority: 3
Risk factor: Some team members are not available full time
Probability: 25%
Impact: One team member will be doing 90% of the work
Mitigation Strategy: Find a good time to meet up with everyone
Monitoring Approach: The team member doing most of the work will have the power to replace any team member who isn't available to work on the project
Management Plan: The team will need to agree on a good time and work together
Estimated resources: All team members

Risk type: Project Size Risk
Priority: 3

Risk factor: Transaction database is not extensible enough to support future additions
Probability: 15%
Impact: The transaction database will be limited
Mitigation Strategy: Create a database with some room for any future additions
Monitoring Approach: If the database isn't extensible, it will be difficult to implement future innovations
Management Plan: The team will make sure the transaction database is future proofed
Estimated resources: Two team members will be responsible for creating the transaction database

Risk type: Project Size Risk
Priority: 3
Risk factor: Downtime caused by lack of storage hardware. (Unplanned scalability)
Probability: 20%
Impact: Upset users due to downtime
Mitigation Strategy: Check storage percentage and add more storage if necessary
Monitoring Approach: Upgrade the storage
Management Plan: Get a hard drive or a solid-state drive with high terabytes
Estimated resources: HDD or SSD

Risk type: User Experience
Priority: 4
Risk factor: User complaints about lack of the latest visual styling standards in web interface
Probability: 10%
Impact: Negative reviews from users
Mitigation Strategy: Allow users to provide feedback on SmartSplit's web interface
Monitoring Approach: Read user's feedback and update the website accordingly
Management Plan: User feedback will be important to improve the website's interface so the team will rely on those opinions
Estimated resources: A team member will be responsible for creating the option to allow users to leave a review on the website

4. Project Schedule

4.1 Project Tasks

- Conception
  - Construct a software requirement specification
    - Review the product criteria
    - Determine requirements
    - Rank requirements in alignment to criteria
- Planning
  - Project estimation
    - Time estimates
    - Labor and misc. Cost estimates
    - Risk estimates
  - Scheduling
    - Decompose functional requirements
    - Develop a task network
    - Develop a timeline for task completions
- Design
  - Analysis
    - Functional analysis
  - UML Diagramming
    - Develop Class Diagrams
- Implementation
  - Write source code
    - Implement Design
  - Testing
    - Unit test
    - Integration test
    - System test
    - Meets requirements

        - Stability test
        - Usability test
- Deployment
    - Deliver
        - Deliver to client
    - Maintain
        - Support
            - Client feedback
        - Update

4.2 Functional Decomposition

Function Name	Description
Account Registration	The account registration shall allow users to create an account. It will require the user to provide their full name, home address, email, and a payment method choice.
Update Account info	The Update Account information function shall allow users to modify their name, mailing/billing address, email, payment methods and password.
Account Login	The account Loginfunction shall allow the user the begin their account session.
Account Logout	The account Logout function shall allow the user the end their account session.
Expense Creation	The expense creation function shall allow users to create an expense. It will require the selection/creation of a group to involve in the expense.
Create Group	The create group function shall allow users to edit the create a group. It will require a name and atleast one other member to be added upon creation.
Edit Group	The edit Group function shall allow users to edit the existing members in a given group.

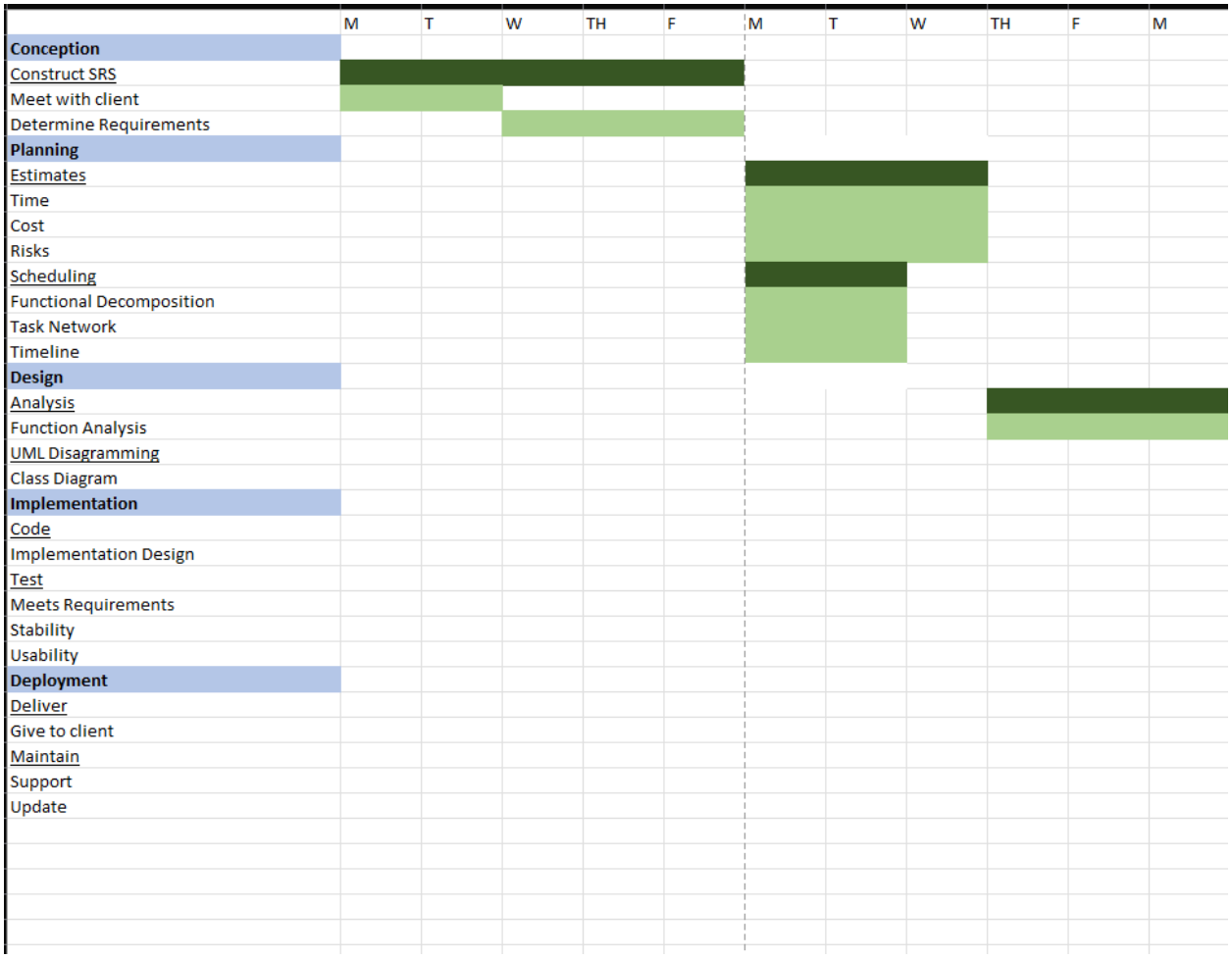
Delete Group	The Delete Group function shall allow users to delete their group if all balances have been settled.
Add Friend	The Add Friend function shall allow the user to add a friend via existing group or email address.
Invite Friend	The invite friend function shall allow the user to invite an existing friend to a group.
Remove Friend	The remove friend function shall allow the user to remove a friend from their list.
Balance Tracking	The balance tracking function shall allow the user to view their outstanding and settled balances.
Notification	The Notification function shall allow the user to receive an email notification from the app.
Transaction History	The Transaction history function shall allow a user to access all transactions involving their account.

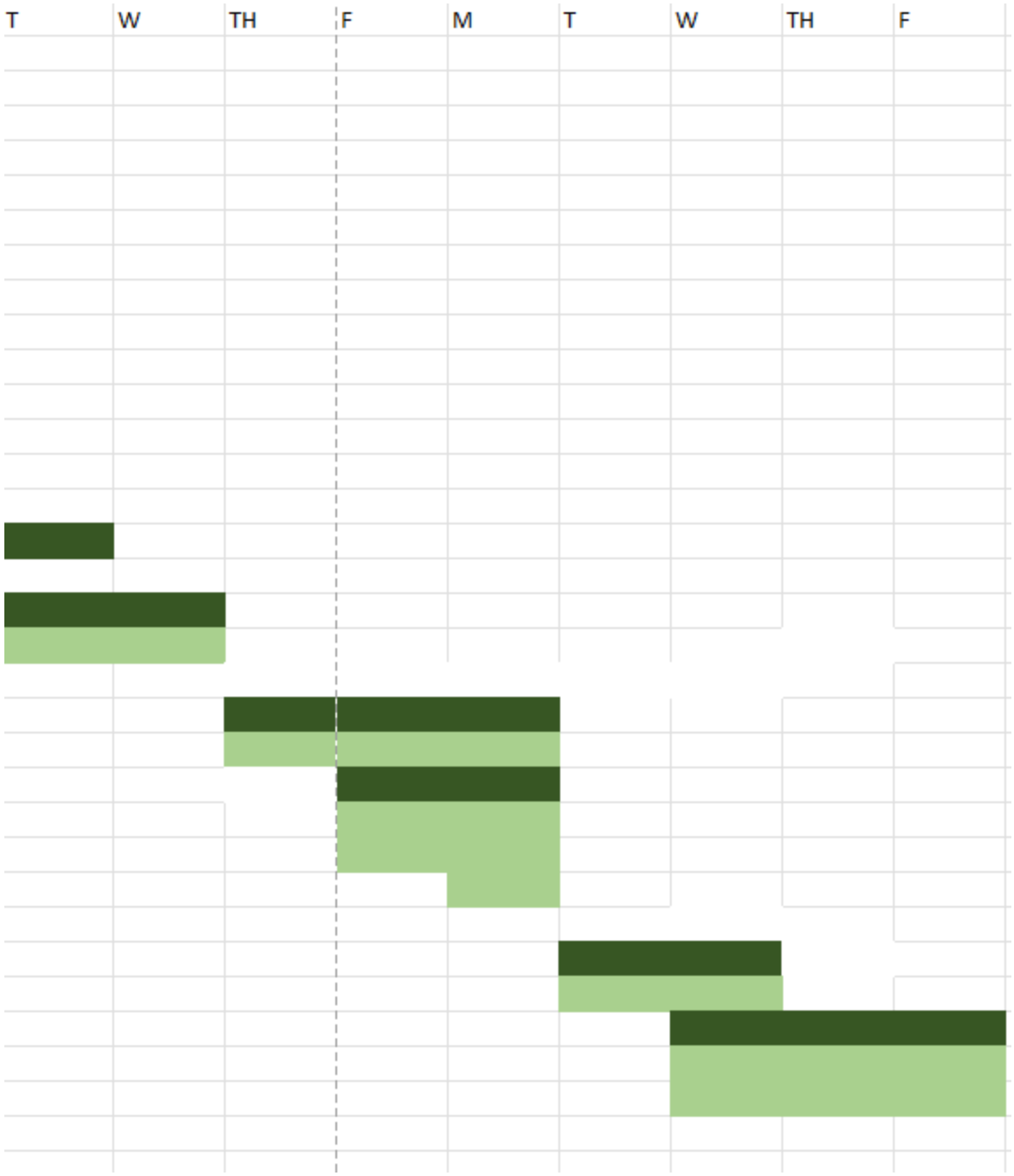
4.3 Task Network

- Conception
  - SRS Planning
  - Review Requirements
- Planning
  - Estimates
    - Time
    - Cost
    - Risks
  - Scheduling
    - Function Decomposition
    - Task Network
    - Timeline
- Design
  - Analysis
    - Function Analysis

    - UML Diagramming
        - Class Diagram
- Implementation
    - Code
        - Implementing Design
    - Test
        - Meets Requirements
        - Stability
        - Usability
        - Portability
- Deployment
    - Deliver
        - Give to Client
    - Maintain
        - Support
        - Update

4.4 Timeline Chart





5. Staff Organization

This section of the document will outline the organization and roles of the members of the software development team.

5.1 Team Structure

None of the team members have expert knowledge in web development or mobile applications. The lack of knowledge does not scare the team, we are motivated to learn and not afraid of any challenge. However, this may mean that the team may have to spend time familiarizing themselves with concepts they may utilize during the development process.

Building SplitSmart will be difficult but not impossible; in order to make sure the team is moving forward, weekly meetings and communication will ensure the project's completion. The team is quite small for this big scale project, so an agile team approach would be a good option. Continuous communication and on-time delivery are stressed in an agile development environment.

The following team roles have been defined:

Documentation: This role will be responsible for requirement analysis and system design as well as organize and manage documentation needed for the software.

- The group member in charge of documentation will ensure that each document contains relevant and consistent information throughout and that documents are completed on time and in an efficient manner.

Design: This role will be responsible for system and program design.

- The group member in charge of design will be responsible for creating innovative unique ideas that can be useful for the system and program. The member will be required to write pseudo code to explain the system and sketch a website to illustrate the program.

Programming: This team will handle program design and implementation.

- The group member in charge of programming will be responsible for directing the software development team in the implementation of the software itself. They will determine the IDE, as well as ensure that the code remains clean and readable throughout.

Testing: This role will provide unit, integration, and system testing for the final program.

- The group member in charge of testing will be responsible for directing the testing phase of the project. They will determine the best method to test as well as lead the creation of test cases.

Training: This role will be responsible for ensuring that all members of the development team understand the concepts we will be working with.

- The group member in charge of training will research topics needed for the creation of the project as well as keep the rest of the team up to date with new concepts they should familiarize themselves with.

Since our team consists of 5 members, each team member will be responsible for one of the roles.

5.2 Management Reporting and Communication

The software development team partakes in daily communication via a WhatsApp group chat to stay up to date on the project and status of group members. Since the group practices the agile development process, there is a scheduled weekly meeting via Zoom

every Wednesday at 7:00pm where each member of the group will discuss the following topics in a scrum-like manner.

1. **What did you do this past week?**
2. **What will you do this week?**
3. **Any impediments you encountered?**

Once each member of the group has given their summary, the team will discuss the next steps needed to move forward. This includes going over the class lecture, as well as setting goals to aim for in the upcoming week. Finally the group will come up with any questions they may have about the project and if there is no clear answer, the group can reach out to Professor Tommy or the TA for clarification.

6. Resources/References

Average Salary Prices:
https://www.salary.com/research/salary/recruiting/web-developer-salary/mi