Splitsmart Term Project

Planning Document

Team 6: Shahd Mustafa, Mike Nasser, Hassan Radwan, Hussen Al-Jubury, Nathaniel Leonardo

1.0 Introduction

The project aims to create a software system “SplitSmart” that allows users to share and track expenses between groups of people allowing users to settle and make payments that are owed. Users can access the software system through a web browser. The system will be capable of allowing users to complete a variety of tasks such as user accounts, group management, expense creation, notification system, balance tracking, payment tracking and reporting. The site could be used to organize and manage a friends/ family trip.

1.1 Project scope

The final software project will have the following features:

Users will be able to:

1. Create an account and view balance information
2. Invite friends/family, etc to join group
3. Manage group expenses and balance
4. Creating new expense containing information such as amount, date, receipt, description and how the expense is split

The software system will be able to:

1. Expense undergo a approval process
2. Notify users when new expenses are created.
3. Track and update users balance update based on expenses made
4. Record payments and settle owed balances
5. Create reports of expenses, balance, and payments
    a. Create a custom report with user desired specifics.

1.2 Major software functions

Required system function that are to be implemented:

- User accounts:
  Users are able to create and access their account and are able to view their balance and expense information.
- Group management:
  Users are able to create, invite and manage groups. Users can track and view shared expenses and balance.
- Expense creation:

Users are able to create new expenses and can specify the amount, date, description, shared manner and receipt options. Users can specify which member of the group the expense is for.
The system will conduct an approval process for any expense made.

- Notification system:
  The system will notify users when new expenses are created by other members of the group.
- Balance tracking:
  The system will track balance owed by all users to others, accounting for all expenses that are made.
- Payment tracking:
  The system will allow all users to record the date of when payments are settled. The system will update to be accurate.
- Reporting:
  The system will generate user reports of all expenses, balance and payments, user can also request a custom report with specific dates.

1.3 Performance/Behavior issues

The software system will have many functionality, the most important feature that will hinder the system performance is responsiveness and security. Users should be able to interact with the user interface easily and actions need to be accurate and quickly done, this includes creating new expenses, balance tracking and payments. Secondly, the system needs to be able to protect user account information such as passwords, bank information, and personal data.implementing security measures by encryption and user authentication.

1.4 Management and technical constraints

The software system is to be fully completed and fully functional in 11 weeks. Members of the team will take 1 week to familiarize themselves further with the chosen software language to speed up the coding process. Making the project have 10 weeks to be fully completed.

2.0 Project Estimates

2.1 Historical data used for estimates

We do not have any historical data to use for estimates from past projects, however we can give the average productivity of 28 fp/person-month with 750 LOC/person-months for the rough estimates to complete the system. The labor rate is $11,000 per month.

2.2 Estimation techniques applied and results

2.2.1 Estimation FPA

External Inputs (EI)

1. User Account creation information(name, email, password)
2. User login credentials(email, password)
3. Expense creation information (amount, date, description, shared manner, receipt options, group/member selection)
4. Group creation information(name, members)
5. Group invitation information(group, recipient)
6. Expense approval information (expense ID, approval status)

External Outputs(EO)

1. User account information (name, email, balance, expense information)
2. Group information (name, members, shared expenses, balances)
3. Expense information (amount, date, description, shared manner, receipt options)
4. Balance information (balance owed by users to others)
5. Payment Information (date of payment settlement)
6. User reports (expenses, balance, payments)
7. Custom reports(expenses, balance, payments for specific dates)

External Inquiries (EQ)

1. Balance inquiry (balance owed by users to others)
2. Expense inquiry (expense details)
3. Group inquiry (group details, shared expenses, balances)
4. Payment inquiry (payment details)

Internal Logical Files (ILF)

1. User Account Files(name, email, password, balance)
2. Expense file(amount, date, description, shared manner, receipt options, group/member details)
3. Group File (name, members, shared expenses, balances)
4. Balance file (balance owed by users to others)
5. Payment file (date of payment settlement)

Information Domain Value	Count	Weight Factor	Total

External Input (EI)	6	4	24
External Output (EO)	7	5	35
External Inquiries (EQ)	4	4	16
Internal Logical Files (ILF)	5	10	50
Total			125

2.2.2 Estimate for LOC-Based Estimation

Estimate generated for LOC

Weighting Factors:
External Inputs: 4
External Outputs: 5
External Inquiries: 4
Internal Logical Files: 10

LOC

Function	FP	Weighting Factor	LOC/FP	Total LOC
Account Creation	EI, ILF	4+10	47	658
View Balance Information	EQ, EO	5+4	47	423
Group Creation	EI, ILF	4+10	47	658
Update Group Information	EI	4	47	188
Add Account to Group	EI	4	47	188
Account Login	EQ	4	47	188
Account Logout	EQ	4	47	188

Create Expense Information	EQ, EO, ILF	5+4+10	47	893
Approve Expense	EQ, EO	5+4	47	423
Update Account Information	EI	4	47	188
View Payments	EQ, EO	5+4	47	423
View Balance	EQ, EO	5+4	47	423
Generate Reports	EQ, EO, ILF	5+4+10	47	893
Total:				5734

2.3 Reconciled Estimate

Estimate for FPA-Based Estimation:
Average Productivity = 28 FP/pm
Burdened Labor Rate = $11,000 per month
Cost per FP = 11,000/28 =$393 per FP

Total Estimated Project Cost: $393x125 = $49125
Total Estimated Effort = 125/28 = 4.46 person-months
Estimated Duration = 4.46/4 = 1.12 months

Estimate for LOC-Based Estimation:
Average Productivity = 750 LOC/pm
Cost per LOC = 11000/750 = 14.67

Total Estimated Project Cost 14.67 x 5734 = $84118
Total Estimated Effort 5734 / 750 = 7.64 person-months
Estimated Duration: 7.64/4 = 1.91 months

Reconciled Estimate:
Final Estimated Cost: $66622
Final Estimated Effort: 6.05 person-months
Final Estimated Duration: 1.51 months

2.4 Project Resources

People: 4 CIS students with minimal experience
Hardware: Personal Computers
Software: html/CSS for interface and JavaScript for website functionality
Tools: Enterprise Architect for UML, Visual Studio and Visual Studio Code for IDE

2.5 project estimates

3.0 Risk Management complete

3.1 Project Risks

1.Insufficient security of login information systems could lead to vulnerability of user information and financials should payment processing be implemented.

2. Incorrect internal calculations could lead to false data provided to the user, resulting in poor feedback

3. Poor UI design could lead to user confusion and frustration resulting in poor feedback.

4. Documentation of expenses and payments could lead to inaccurate reporting.

5. Poor communication among team members and clients could lead to a hasty or poorly managed schedule resulting in bad time management.

6.Poor scheduling amongst development team leading to rushed deadlines and inappropriate distribution of time

3.2 Risk Table

The complete risk table is presented. Name of risk, probability, impact and RM3 pointer are provided.

Risk	Impact	Probability	RM3
Security	Very High	60%	*

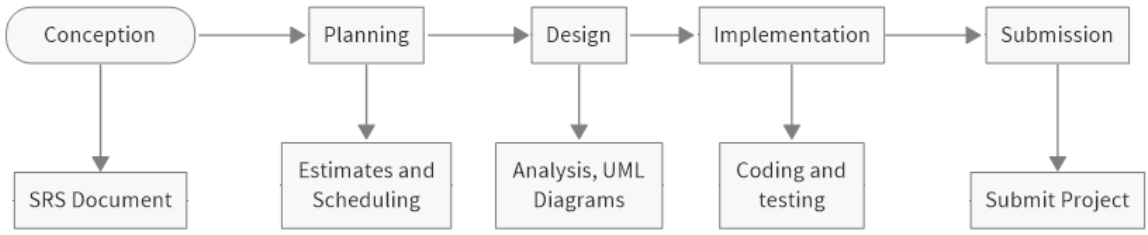
Poor scheduling	High	30%	**
Reporting issues	Medium	10%	***
Calculation Accuracy	Medium	20%	****
Group communication Issues	Medium	40%	*****
Changes in Requirements	Low	20%	******

3.3 Overview of Risk Mitigation, Monitoring, Management

RM3 Pointer	Mitigation	Monitoring	Mgmt
*	Reviewing security measures during development with ample testing	Frequent testing throughout stages of development	Communication and review amongst group to accommodate for any changes
**	Weekly team meetings and communication between team members and clients	Checking on progress on the project regularly in meetings	Adapting the planned schedule to accommodate any issues found
***	Thorough checks on calculation formulas and results	Monitoring and checking report calculations and results	Correcting and retesting calculations
****	Thorough checks on calculation formulas and results	Monitoring and checking calculations within each group and event	Correcting calculation formulas and retesting calculations
*****	Weekly meetings and a shared Github	General progress on the project as well as	Weekly meetings will improve communications

	page will facilitate communication	functionality of the program	
******	Being flexible with the project plan and being communicative with the client	Keeping track of requirements and confirming aspects with the client	Adapting the project plan to accommodate changes

4.0 Project Schedule (Mike)



4.1 Project task set

Conception

- Software Requirements Specification (SRS)
- Determine Requirements

Planning

- Estimates
    - Time
    - Cost
    - Risks
- Scheduling
    - Functional Decomposition
    - Timeline
    - Task Outline

Design

- Analysis
    - Function Analysis
- UML Diagrams
    - Class Diagrams

Implementation

- Functional Code
- HTML Design
- Testing
    - Usability
    - Stability
    - Meets Requirements

Submission

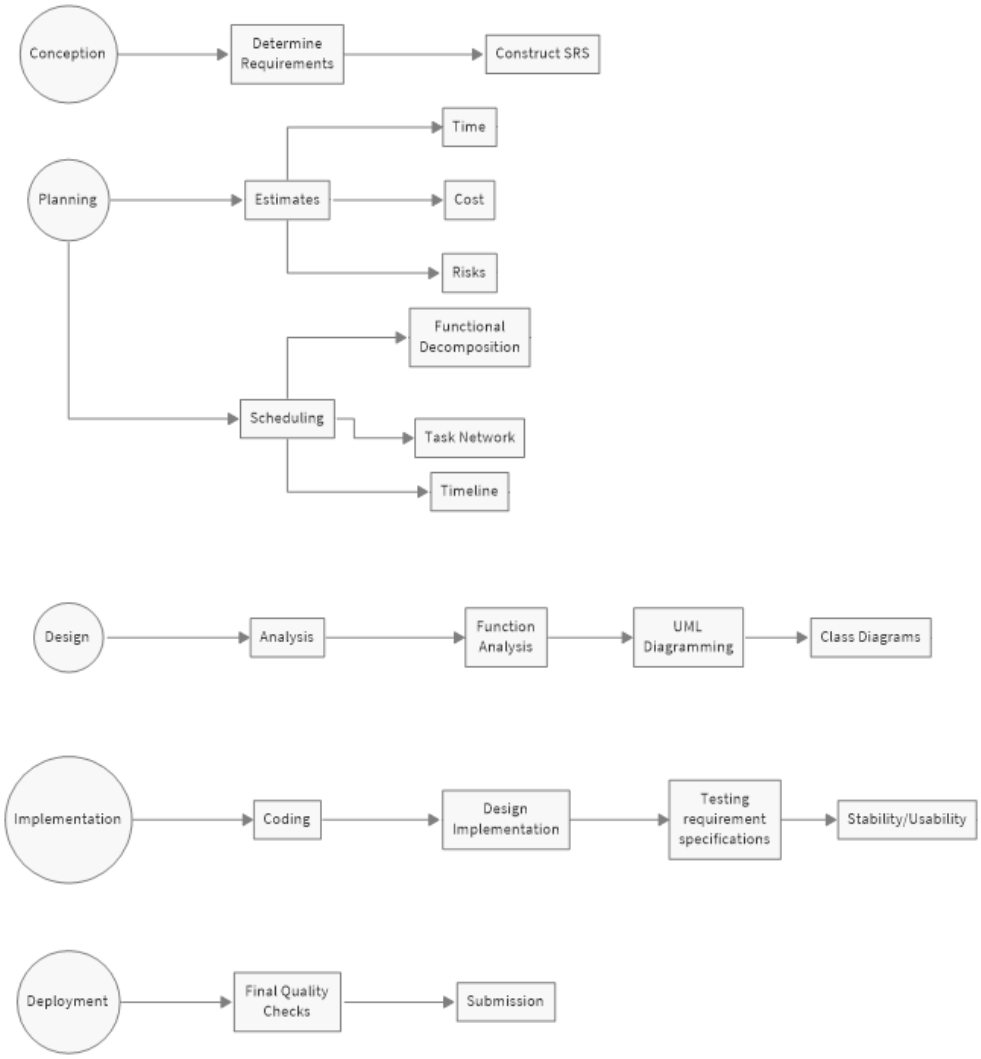
- Deliver
    - Submit to client
- Maintain
    - Support/Update as needed

4.2 Functional decomposition

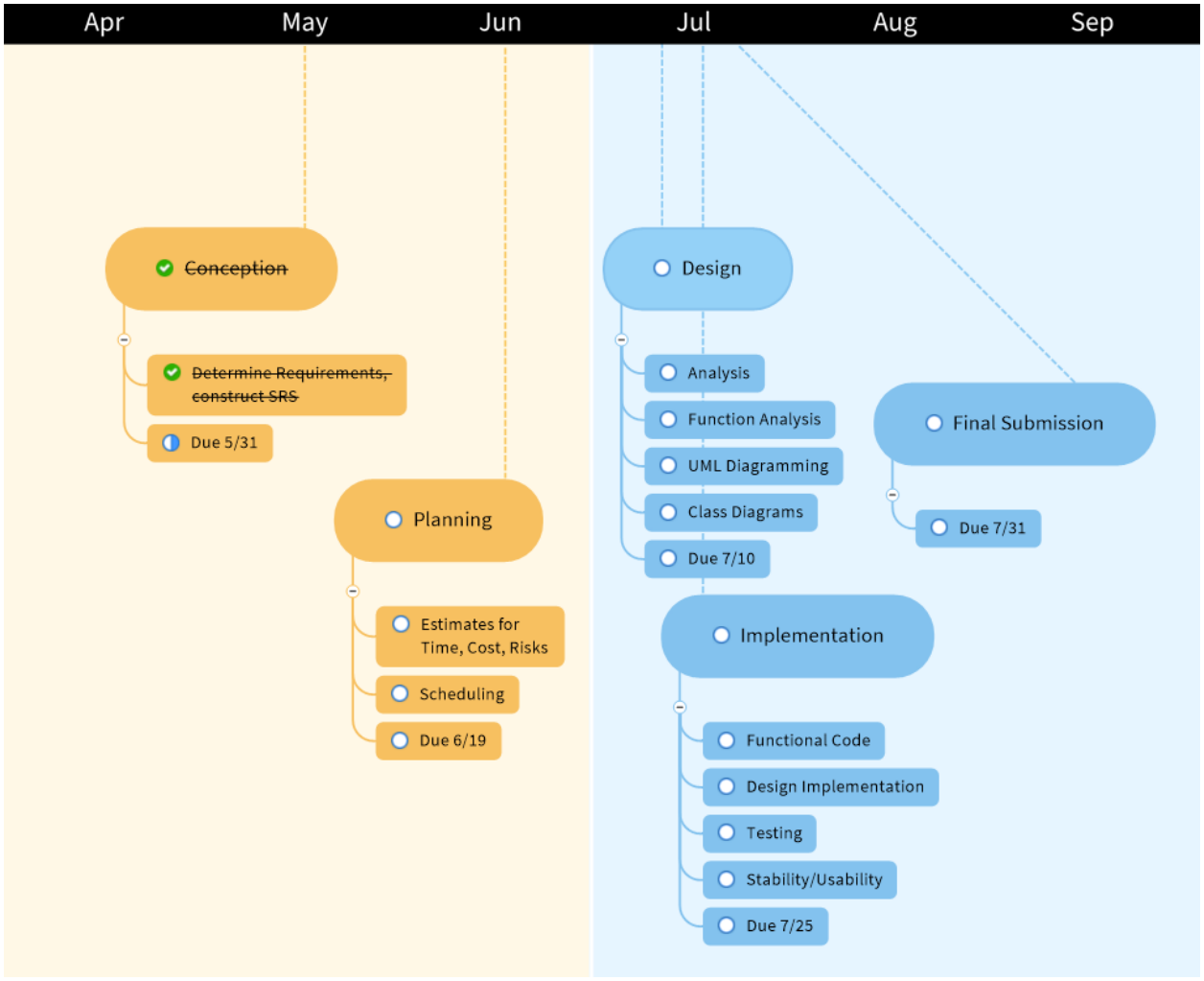
Function Name	**Description**
Account Registration	**Registration functions should allow users to create accounts. Users setting up an account should provide name, address, billing address, Credit card type, card number, expiration date and desired username/password**
Login	**Login function should prompt user for username/password and once verified should allow user access to their account**
Add Expense	**The Add Expense function should allow users to add a bill to their dashboard. Should require a bill amount and a description. This function should prompt the user to designate who the expense is being shared with, and the split**

	distribution. Should prompt for the date of expense and give the option to add images/notes.
Add Group	**Should allow users to give the group a name and invite others via name and email address to the group.**
Payment	**The payment function should prompt for a payment method. Should provide options to record a cash payment, use venmo, paypal, etc.. Function should record when payments have been made so that balances update accurately. A real payment will not be required for this project, should a button that changes the balance.**
Notify	**The notify function should alert users when an expense has been created in their group**
Balance	**The balance function should track the amount owed by each user and who they owe.**
Report	**Report function should give users a summary of their expenses, balances, and payments.**

4.3 Task network



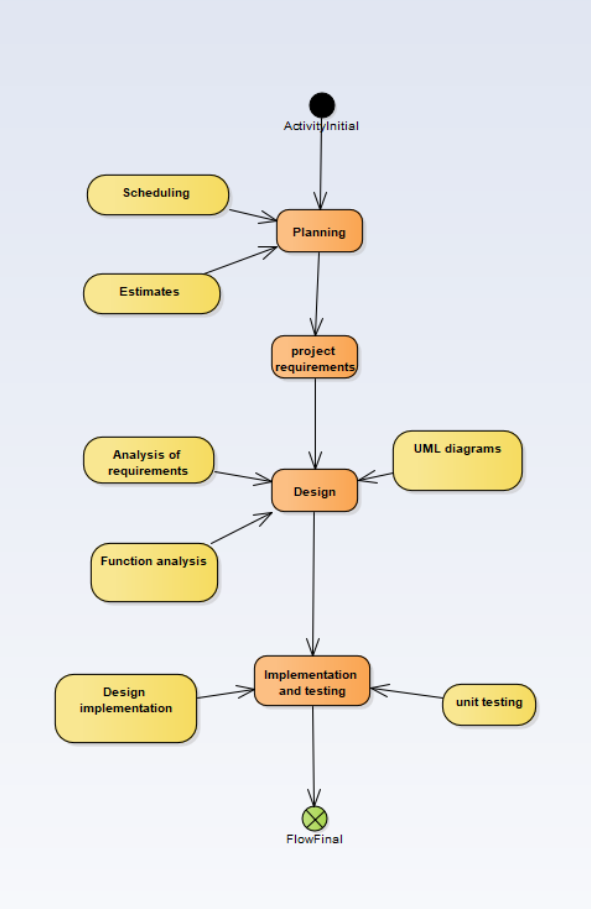
4.4 Timeline chart/Product Roadmap



Task Name	Duration	Start	Finish	Predecessors	Resource Names
Software Development	**61 days**	**Mon 5/8/23**	**Mon 7/31/23**		
Scope	**18 days**	**Mon 5/8/23**	**Wed 5/31/23**		
Determine project scope	**1 hr**	**Mon 5/8/23**	**Mon 5/8/23**		**Project leader**
project estimates	**0.25 days**	**Mon 5/8/23**	**Mon 5/8/23**	**2**	**Project leader**
project scheduling	**0.25 days**	**Mon 5/8/23**	**Mon 5/8/23**	**3**	**Designer**
risk management	**0.13 days**	**Mon 5/8/23**	**Mon 5/8/23**	**4**	**Designer**
Scope complete	**17.25 days**	**Mon 5/8/23**	**Wed 5/31/23**	**5**	
Analysis/Software Requirements	**13 days**	**Thu 6/1/23**	**Mon 6/19/23**		
Design	**15 days**	**Tue 6/20/23**	**Mon 7/10/23**		
Development	**58 days**	**Mon 5/8/23**	**Wed 7/26/23**		
Code	**1.63 days**	**Tue 7/11/23**	**Wed 7/12/23**	**22**	**Developer**
implement design	**1.13 days**	**Wed 7/12/23**	**Thu 7/13/23**	**24**	**Developer**
Developer testing (primary debugging)	**0.38 days**	**Mon 5/8/23**	**Mon 5/8/23**		**Developer / Tester**
Development complete	**0 days**	**Wed 7/26/23**	**Wed 7/26/23**	**26**	

Testing	**3 days**	**Thu 7/27/23**	**Mon 7/31/23**		
Integration Testing	**1.75 days**	**Thu 7/27/23**	**Fri 7/28/23**		
Test module integration	**0.5 days**	**Thu 7/27/23**	**Thu 7/27/23**		**Developer / Tester**
Identify anomalies to specifications	**0.38 days**	**Thu 7/27/23**	**Thu 7/27/23**	**30**	**Developer / Tester**
Modify code	**0.5 days**	**Thu 7/27/23**	**Fri 7/28/23**	**31**	**Developer / Tester**
Re-test modified code	**0.38 days**	**Fri 7/28/23**	**Fri 7/28/23**	**32**	**Developer / Tester**
Integration testing complete	**0 days**	**Fri 7/28/23**	**Fri 7/28/23**	**33**	
Software development template complete	**0 days**	**Mon 5/8/23**	**Mon 5/8/23**		

Activity diagram:



5.0 Staff Organization -complete

Our staff follows an agile team structure.

5.1 Team structure

Given that our team consists of only five members, we have determined that an agile team structure is the most suitable approach for us. While our collective experience in web programming is limited, we recognize the importance of working collaboratively and utilizing our skills and resources effectively to deliver high-quality code. Our primary focus is establishing clear channels of communication among team members to foster effective collaboration.

To facilitate our project's success, we have assigned the following team roles:

Team Lead (one person): The Team Lead is responsible for ensuring that the team remains on track and adheres to the established schedule. Additionally, they will host weekly Zoom meetings and diligently document meeting notes for the entire team.

Designer (one person): The Designer's role involves analyzing the client's requirements and developing a website design that best meets their needs. As the project progresses, the Designer will continuously refine the design to ensure exceptional quality of work.

Developers (three people): The Developers are responsible for the creation and implementation of the software. They must maintain constant communication with one another, as well as with the Team Lead and Designer, to ensure a smooth and efficient development process. The Developers are also accountable for conducting thorough unit, integration, and system testing to ensure the software's robustness.

5.2 Management reporting and communication

We have established a compulsory weekly virtual gathering on the Zoom platform every Friday afternoon. These meetings serve as a means to provide comprehensive updates on individual progress throughout the week, as well as to determine the tasks and objectives for each team member in the upcoming week. The primary focus of these meetings revolves around sharing significant updates and assigning key responsibilities. Notes of these meetings will also be committed to our shared Github repository to ensure that in the unlikely event that a member misses an event we still ensure they are up to date.

Furthermore, we have implemented a dedicated group chat on the Whatsapp messaging platform. We can quickly resolve any unforeseen changes or problems that may develop thanks to the group chat's function as a medium for speedy and direct communication between team members. We provide a constant flow of contact among team members by combining the weekly meetings and the Whatsapp group conversations, efficiently resolving both minor and significant concerns.