**CS3354 Software Engineering**

**Final Project Deliverable 2**

**Split Squad**

By: Tan Vo, Daniel Nguyen, Stephen George, Quan Pham,

Mir Ali, Samuel Preston, Amaan Babul, Shriniketh Mukundan

**GitHub Repository**: <https://github.com/md-y/3354-splitsquad>

**1. [5 POINTS]** Well described delegation of tasks, i.e. who did what in the project. Now that your project is complete, you are required to submit the delegation of tasks from beginning of the project until the end. Please make sure to fairly distribute tasks in the team and remember that in the end of the semester, each member of a team will receive the same grade. See grading policy below for more detail. If no/poor contribution by a member, please specify clearly so that we can grade each student fairly.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Description** | **Task Owner** | **Task Description** | **Task Owner** |
| GitHub Invitation | Samuel Preston | Project Scheduling | Daniel Nguyen |
| README | Tan Vo | Cost, Effort, Pricing | Shriniketh Mukundan |
| Project Scope | Stephen George | Hardware Cost | Stephen George |
| Process Model | Quan Pham | Software Cost | Tan Vo |
| Functional Requirements | Daniel Nguyen | Personnel Cost | Quan Pham |
| Nonfunctional Requirements | Mir Ali | Unit Testing | Samuel Preston |
| Use case Diagram | Samuel Preston | Comparison | Mir Ali, Amaan Babul |
| Sequence Diagram | Tan Vo |  |  |
| Class Diagram | Shriniketh Mukundan |  |  |
| Architecture Design | Amaan Babul |  |  |

**2. [10 POINTS]** Everything required and already submitted in Final Project Deliverable 1. Please specify this part as “Project Deliverable 1 content”.

*IMPORTANT NOTE: The following items will all need to be calculated / worked on based on the project you are designing. As an example, if a team of 7 students in CS3354 class is working on the development of a hospital information system, this group will prepare the project scheduling, cost, effort and pricing estimation calculations based on the hospital information system design, NOT based on their 7 student team. Think of the analogy to the “Inception” movie: What you will be working on is the dream in a dream, i.e. the dream in the second level, NOT in the first level.*

**3. [35 POINTS] Project Scheduling, Cost, Effort and Pricing Estimation, Project duration and staffing:** Include a detailed study of project scheduling, cost and pricing estimation for your project. Please include the following for scheduling and estimation studies:

**3.1. [5 POINTS] Project Scheduling.** Make an estimation on the schedule of your project. Please provide start date, end date by giving justifications about your estimation. Also provide the details for:

- Whether weekends will be counted in your schedule or not

- What is the number of working hours per day for the project

|  |
| --- |
|  |

**3.2. [15 POINTS] Cost, Effort and Pricing Estimation**. Describe in detail which method you use to calculate the estimated cost and in turn the price for your project. Please choose one of the two alternative cost modeling techniques and apply that only:

- Function Point (FP)

- Application composition

We all decided that using the FP model would be the algorithmic technique for our group project itself due to the fact that there are many benefits when pertaining to implementing this method. For example, we thought this method would be useful since this method has a very standard approach that helps to address all the different factors including the inputs, outputs, and other aspects of the program that contribute to the overall functionality of the project. Additionally, we thought that the FP method would be really useful from a scalability perspective due to the fact that there are many instances during the duration of projects where new resources need to be added for enhancing the overall structure and functionality of the system and as a result, this method would be really useful when it came to understanding how those changes can impact the overall time needed for the project along with the change in price and cost.

Number of user input:

- Login Information (Username/Email & Password)

- Add Friend (Username/Email Information)

- Remove Friend (Username/Email Information)

- Update Friend Request Status (Accept/Decline)

- Add Group (Group Name & Initial Members)

- Add Friend to Group (Group Name & Username/Email)

- Delete Group (Group Name w/ Authority)

- Create Split Request (Split Info & Members’ Usernames/Emails)

- Update Split Request (Split Info & Accept/Decline)

- Pay Split Request (Split Info & Payment Info)

- Pay Friend Directly (Member Username/Email & Payment Info)

- Forgive Expense (Username/Email)

- Update Account Settings (New Account Settings)

 Number of user output:

- Friend Request Notification

- Group Invite Notification

- Split Request Notification

- Payment Notification

-Friends view

-Profile/Settings View

-Transaction History View

-Split Creation View

-Group View

Number of user queries:

- Search Friends

 - List All Friends

- List All Friend Requests

- List All Joined Groups

- List All Split Requests

- View Expense Statistics

- View Personal Profile

- View Transaction History

Number of data files and relational tables:

- User Account Information:

-Friends

- Payment Methods

- Split Requests:

-Split Transactions

-Split Members

- Groups

- Group Members

- System Configurations

- Encryption Keys

 Number of external interfaces:

- Payment-to-Payment System

- Verify Balance w/ Payment System

- Verify New Payment Method with Payment System

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Function Category | Count | Complexity | Complexity | Complexity | Count \* Complexity |
|  |  |  | Simple | Average | Complex |  |
| 1 | User Inputs | 13 | 3 | 4 | 6 | 52 |
| 2 | User Outputs | 9 | 4 | 5 | 7 | 45 |
| 3 | User Inquiries | 8 | 3 | 4 | 6 | 32 |
| 4 | Data Files and Relational Tables | 10 | 7 | 10 | 15 | 150 |
| 5 | External Interfaces | 3 | 5 | 7 | 10 | 30 |

GFP=52+45+32+150+30=309

PC1=5

PC2=3

PC3=3

PC4=3

PC5=2

PC6=5

PC7=5

PC8=3

PC9=3

PC10=5

PC11=4

PC12=3

PC13=4

PC14=4

Total= 52

PCA=0.65+(0.01\*52) = 1.17

FP=1.17\*309= 361.53

Assuming there’s 40 FP/person weeks  in terms of productivity:

=> 361.53/40=9.03825 person weeks

Cost Estimation:

9.038 Weeks=2.08 Months

Total Hardware Cost: $469.27\*2.08= $976.08

Total Software Cost: ($547.5\*2.08)+25= $1163.8

Total Personnel Cost: ($67066.67\*2.08)+100= $139,598.67

Total Overall Project Cost: $141,738.55

Selling Price Per User:

Total number of Users: 170K

Break Even Cost: $141,738.55/170,000= $0.834

If we are trying to have a profit margin of 20%, the corresponding selling price per user would be 0.834\*1.2=$1 per user.

|  |
| --- |
|  |

**3.3. [5 POINTS]** Estimated cost of hardware products (such as servers

|  |  |
| --- | --- |
| **Cloud Services** | **Cost** |
| **Azure Cloud Services Virtual Machine (with IIS Web Server)**  A2m v2 processor (2 Cores, 16 GB RAM): $0.195/hour []  Considering a 730 hour month: (730)(0.195) = $142.35/month | $142.35 / month |
| **Database** |  |
| **Azure SQL Database**  Standard-Series (Gen 5) Single Database, vCore model (2):   $0.36532/hour and $0.138/GB []  Considering a 730 hour month: (730)(0.36532) = $266.68/month  Considering the size of the logs to be about 30%: (0.138)**(**128**)**(1.3) **= $**22.96/month | $289.64 / month |
| **Database Backup**  Monthly Long Term Backup Retention: $0.06/GB/month []  Backing up the entire database: (128)(0.06) = $7.68/month | $7.68 / month |
| **Security** |  |
| **Microsoft Defender for Cloud**  Server Plan 2: $0.02/hour []  Azure SQL Database Service: $15/server/month []  Considering a 730 hour month: (730)(0.02) + (1)(15) = $29.60/month | $29.60 / month |
| **Total Cost:** $469.27/month |  |

**3.4. [5 POINTS]** Estimated cost of software products (such as licensed software, etc.)

|  |  |
| --- | --- |
| **Development Tools** | **Cost** |
| **Visual Studio Code Professional (include Azure DevOps):** $45/months/users. We consider a 10 person-team, which allowed for a discount, with the price listed on the right [AA] | $438.75 / month |
| **Third-Party APIs** |  |
| **Stripe**: Stripe is free to set up and the company does not charge a monthly or annual fee for its services. [AB] | $0 |
| **Microsoft Notification Hubs**: The first 10 million pushes are included for 200,000 active devices, which we can assume our apps have the user base below that range. Thus, we only need to pay the default rate of $10/month for the Basic plan. [AC] | $10.00 / month |
| **Testing Tools** |  |
| **Vite & Vitest**: Free | $0 |
| **Playwright**: $0.01/minute testing on Linux. We can roughly assume approximately one test for one function point. Thus, a function points score of 361.53 roughly translated to 362 tests in the test suite. We can also assume roughly one second per test. Given that we have 10 developers who will be running a test suite roughly 5 per day, we will run roughly 1500 test runs per month. In total, Microsoft Playwright Testing will incurs $90.50/month [AC]. | $90.50 / month |
| **App Store Fees** |  |
| **Apple App Store**: $99/year. [AD] | $8.25 / month |
| **Google Play Store**: $25 one-time registration fee [AD] | $25 |
| **Total Cost:** $547.5/month + $25 one-time fee |  |

**3.5. [5 POINTS]** Estimated cost of personnel (number of people to code the end product, training cost after installation)

|  |  |
| --- | --- |
| **Premises** |  |
| 1. The development team consists of 10 members, all of whom are developers with nominal experience. 2. Estimated salary for a developer with nominal experience is $80,000/year (~$6,667/month or ~$1,539/week). 3. All members of the development team are experienced in Visual Studio Code. |  |
| **Training** | **Cost** |
| **Visual Studio Code:** Since all members are experienced in this tool, no training is necessary. | $0 |
| **Azure DevOps:** A training path is provided by Microsoft, and it’s free. No advanced experience is truly required, so a beginner’s course is sufficient []. | $0 |
| **Vite & Playwright:** A course guide for Vite and Playwright can be offered at udemy.com [] []. Using the team plan that the website offers, the team can afford each of these courses with each member being billed at a rate of $30/month []. | $300 / month |
| **Vitest:** A comprehensive guide is already provided on the official website, so minimal training is required (we’ll assume an extra total $100 to cover any additional resources needed) []. | $100 |
| **Achievement Rewards** |  |
| **Employee of the Month:** A team member is recognized for their outstanding work each month and rewarded with a bonus! | $100 / month |
| **Total Cost:** $67066.67/month + $100 |  |

**4. [10 POINTS]** A test plan for your software: Describe the test plan for testing minimum one unit of your software. As an evidence, write a code for one unit (a method for example) of your software in a programming language of your choice, then use an automated testing tool (such as JUnit for a Java unit) to test your unit and present results. Clearly define what test case(s) are provided for testing purposes and what results are obtained (Ch 8). Include your test code as additional document in your zip file submitted.

|  |
| --- |
|  |

**5. [10 POINTS]** Comparison of your work with similar designs. This step requires a thorough search in the field of your project domain. Please cite any references you make.

|  |
| --- |
| In the realm of cost-sharing apps, our platform stands out as a streamlined facilitator for splitting expenses among groups, particularly focusing on meals and various group purchases. When we look at existing apps in this field, there are a few that come close to what we offer. Venmo, for example, lets users split bills and expenses, but it's more geared toward individual transactions rather than group purchases [] []. Split wise is another app that's commonly used for splitting expenses among groups []. However, it doesn't have a direct payment feature and is more of an expense tracker. It's great for keeping tabs on shared costs but doesn't handle payments directly. []  Our app, on the other hand, merges the features of splitting costs among groups and facilitating direct payments, providing a comprehensive solution for communal spending. This sets us apart by offering a more user-friendly interface and efficient payment processing, which isn't fully covered by the existing options available. |

**6. [10 POINTS]** Conclusion - Please make an evaluation of your work, describe any changes that you needed to make (if any), if things have deviated from what you had originally planned for and try to give justification for such changes.

|  |
| --- |
|  |

**7. [5 POINTS]** References: Please include properly cited references in IEEE paper referencing format. Please review the IEEE referencing format document at the URL: [https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf](https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%2520Citation%2520Guidelines.pdf)). It means that your references should be numbered, and these numbers properly cited in your project report.

|  |
| --- |
| [] Microsoft, “Pricing-Microsoft Defender: Microsoft Azure,” Pricing-Microsoft Defender | Microsoft Azure, https://azure.microsoft.com/en-us/pricing/details/defender-for-cloud/ [Accessed Nov. 16, 2023].  [] Microsoft, “Pricing - Azure SQL Database Single Database: Microsoft Azure,” Pricing - Azure SQL Database Single Database | Microsoft Azure, https://azure.microsoft.com/en-us/pricing/details/azure-sql-database/single/#pricing [Accessed Nov. 16, 2023].  [] Microsoft, “Pricing - Cloud Services: Microsoft Azure,” Pricing - Cloud Services | Microsoft Azure, https://azure.microsoft.com/en-us/pricing/details/cloud-services/ [Accessed Nov. 16, 2023].  [AA] Microsoft, “Visual Studio Professional - monthly subscription - Visual Studio Marketplace,” *marketplace.visualstudio.com*. https://marketplace.visualstudio.com/items?itemName=ms.vs-professional-monthly#pricing (accessed Nov. 16, 2023).  [AB] J. Simonson and K. Main, “Stripe Pricing And Fees (2023 Guide) – Forbes Advisor,” *www.forbes.com*, 2023. https://www.forbes.com/advisor/business/services/stripe-pricing-fees/ (accessed Nov. 16, 2023).  [AC] Microsoft, “Pricing Calculator | Microsoft Azure,” *Microsoft.com*, 2023. https://azure.microsoft.com/en-us/pricing/calculator/ (accessed Nov. 16, 2023).  [AD] Swing2App, “How much does it cost to publish an app on the app store? | Blog,” *Swing2App*, Aug. 18, 2022. https://www.swing2app.com/blog/how-much-does-it-cost-to-publish-an-app-on-the-app-store/ (accessed Nov. 16, 2023).  [] “Pricing calculator: Microsoft Azure,” Pricing Calculator | Microsoft Azure, https://azure.microsoft.com/en-us/pricing/calculator/?service=playwright-testing [Accessed Nov. 15, 2023].  [] Chcomley, “Get started with Azure DevOps,” Learning path - Training | Microsoft Learn, https://learn.microsoft.com/en-us/training/paths/evolve-your-devops-practices/ [Accessed Nov. 16, 2023].  [] J. Charlesworth, “Vite: The Complete Developer’s Guide,” Vite: The Complete Developer’s Guide | Udemy, https://www.udemy.com/course/vite-developers-guide/ [Accessed Nov. 16, 2023].  [] B. Alam, “Web Automation and Testing using Playwright,” Web Automation and Testing using Playwright | Udemy, https://www.udemy.com/course/playwright/ [Accessed Nov. 16, 2023].  [] “Scalable learning for organizations of every size,” Plans - Udemy Business, https://business.udemy.com/plans/ [Accessed Nov. 16, 2023].  [] Vladimir, “Getting Started,” Getting Started | Guide | Vitest, https://vitest.dev/guide/ [Accessed Nov. 16, 2023].  [] What Is Venmo? – Venmo, help.venmo.com/hc/en-us/articles/221011388-What-is-Venmo-. [Accessed 15 Nov. 2023].  [] “What Is Venmo: How It Works.” Money, money.com/what-is-venmo-how-it-works/. [Accessed 15 Nov. 2023].  [] “About.” Splitwise, www.splitwise.com/about. [Accessed 15 Nov. 2023].  [] Reiff, Nathan. “How Splitwise Makes Money.” Investopedia, Investopedia, www.investopedia.com/articles/company-insights/090816/how-splitwise-works-and-makes-money.asp. [Accessed 15 Nov. 2023]. |

**8. [10 POINTS]** Presentation slides. No min/max number of slides enforced. Please make sure that you can complete presentation within 20 (twenty) minutes. Following template could be a good start to prepare your presentations. As each project topic is different, a variety in presentation style is expected and welcome.

- Title of your project together with participants

- Objective of the project designed

- Cost estimation

- Project timeline (timeline of the project designed, NOT the time you’ve spent on it)

- Functional and non-functional requirements. If too long, select representative items.

- Use case diagram

- Sequence diagram for a selected representative operation of the project.

- Class diagram

- Architectural design

- Model-View-Controller (MVC) pattern (similar to Figure 6.6)

- Layered architecture pattern (similar to Figure 6.9)

- Repository architecture pattern (similar to Figure 6.11)

- Client-server architecture pattern (similar to Figure 6.13)

- Pipe and filter architecture pattern (similar to Figure 6.15)

- Preferably a demo of user interface design that shows screen to screen transitions though no full functionality is required.

- OPTIONAL: IF implemented the project, a demo of your implementation.

**9. OPTIONAL PART [POSSIBLE EXTRA CREDIT UP TO 10 POINTS].** Your program code (if fully implemented the project, not required otherwise). Please note that implementation is not required for the final project. Groups are welcome to implement their work, if they choose to do so. [This part may qualify for extra credit, if you implement and submit the implementation code together with your project. The extra credit will be determined based on the quality of your implementation. Furthermore, any fully implemented project qualifies for scholar publication afterwards. This most probably will involve further commitment to work more an write a scholar paper to send to a Conference for publication.

**10. [5 POINTS] GitHub requirement:**

Make sure at least one member of your group commits everything for project deliverable 2 to your GitHub repository, i.e.

- Your final project deliverable2 report

- Unit test code for a sample unit of your project

- Implementation code (if you have implemented your project)

- Presentation slides

Still, one member of your team should also submit the required project deliverable 2 materials to eLearning.