

CS 3354 Software Engineering

Final Project Deliverable 2

Title: SMS Messenger Software

Group Members

1. Tommy Nguyen
2. Risheel Gunda
3. Taylor Cheung
4. Radhika Aggarwal
5. Aalhad Pathare
6. Chhay Lay Heng
7. Sofia Trevino

1. Task Delegation - update

Tommy Nguyen - Github setup, software test plan, Unit testing

Risheel Gunda - Response to proposal feedback, software model, software costs, conclusion slide

Taylor Cheung: GitHub setup, software model, Diagram Creation, Personnel cost, overall software cost

Radhika Aggarwal - Report type-up, requirement analysis, unit testing

Aalhad Pathare - GitHub setup, cost estimation, architectural design, hardware costs

Chhay Lay Heng - Presentation slides, project scheduling, comparison and objective slides/portion.

Sofia Trevino - software requirements, class diagram, cost estimation, references, project scheduling

2. Project Deliverable 1 content

Final Project Deliverable 1

10/16/23

Final Project Proposal

Title: SMS Messenger Software

Group Members:

1. Tommy Nguyen
2. Risheel Gunda
3. Taylor Cheung
4. Radhika Aggarwal
5. Aalhad Pathare
6. Chhay Lay Heng
7. Sofia Trevino

Software Idea:

Our group will be designing a SMS Messaging Software for students in order to allow them to easily communicate with other students to organize events, send direct messages, and meet other peers.

Our motivation for this idea and potential uses in real-life:

Our motivation for this project is to aid students in classroom and group settings to promote collaboration and communication between one another. Creating a platform for students to interact in an increasingly digital world allows communities to foster around courses, with peers helping each other out and productivity improving on a large scale.

Task Delegation:

Tommy Nguyen - Github setup, software test plan
Risheel Gunda - Response to proposal feedback, software model
Taylor Cheung: GitHub setup, software model, Diagram Creation,
Radhika Aggarwal - Report type-up, requirement analysis
Aalhad Pathare - GitHub setup, cost estimation, architectural design
Chhay Lay Heng - Presentation slides, project scheduling
Sofia Trevino - software requirements, diagram, cost estimation, references

*Presentation: will divide up based on what topics people covered during the project duration.

Are you interested in writing a scholar paper in the end (no extra grade advantage): Our group is not interested in writing a scholar paper.

Feedback:

A lovely topic!! Once complete, it will save a lot of time and effort for those who are to communicate for collaboration in school. Please consider implementing it fully, if you can. No pressure w.r.t. grade on implementation. In the final report, please make sure to include

comparisons with similar applications -if any-, make sure that you differentiate your design from those, and explicitly specify how. Fair delegation of tasks. Please share this feedback with your group members. You are good to go. Have fun with the project and hope everyone enjoys the collaboration.

Response to Feedback:

We plan to incorporate the majority of the changes in the Final Report by providing similar software to ours and what in particular sets ours apart from existing softwares. We plan to research similar softwares such as GroupMe, Discord, etc. and find ways in which we differ in order to list the differences in the final report.

1. GitHub Setup

Repository link: <https://github.com/radhikaaggarwal19/3354-Messenger-Software>

2. Task Delegation:

Tommy Nguyen - Github setup, software test plan

Risheel Gunda - Response to proposal feedback, software model

Taylor Cheung: GitHub setup, software model, Diagram Creation,

Radhika Aggarwal - Report type-up, requirement analysis

Aalhad Pathare - GitHub setup, cost estimation, architectural design

Chhay Lay Heng - Presentation slides, project scheduling

Sofia Trevino - software requirements, diagram, cost estimation, references

3. Software Model - pick one and detail a little about what and why

- a. Our group has decided on doing the incremental process model since we believe it best suits our needs for the SMS software. The incremental model will allow us to have early delivery with partial functionality of our product while being able to still go back and modify any aspects if necessary after each incremental delivery. In addition to that the incremental process promotes reduced risk and easier testing since we are not fully committing to the final product in a single process run.

4. Software requirements

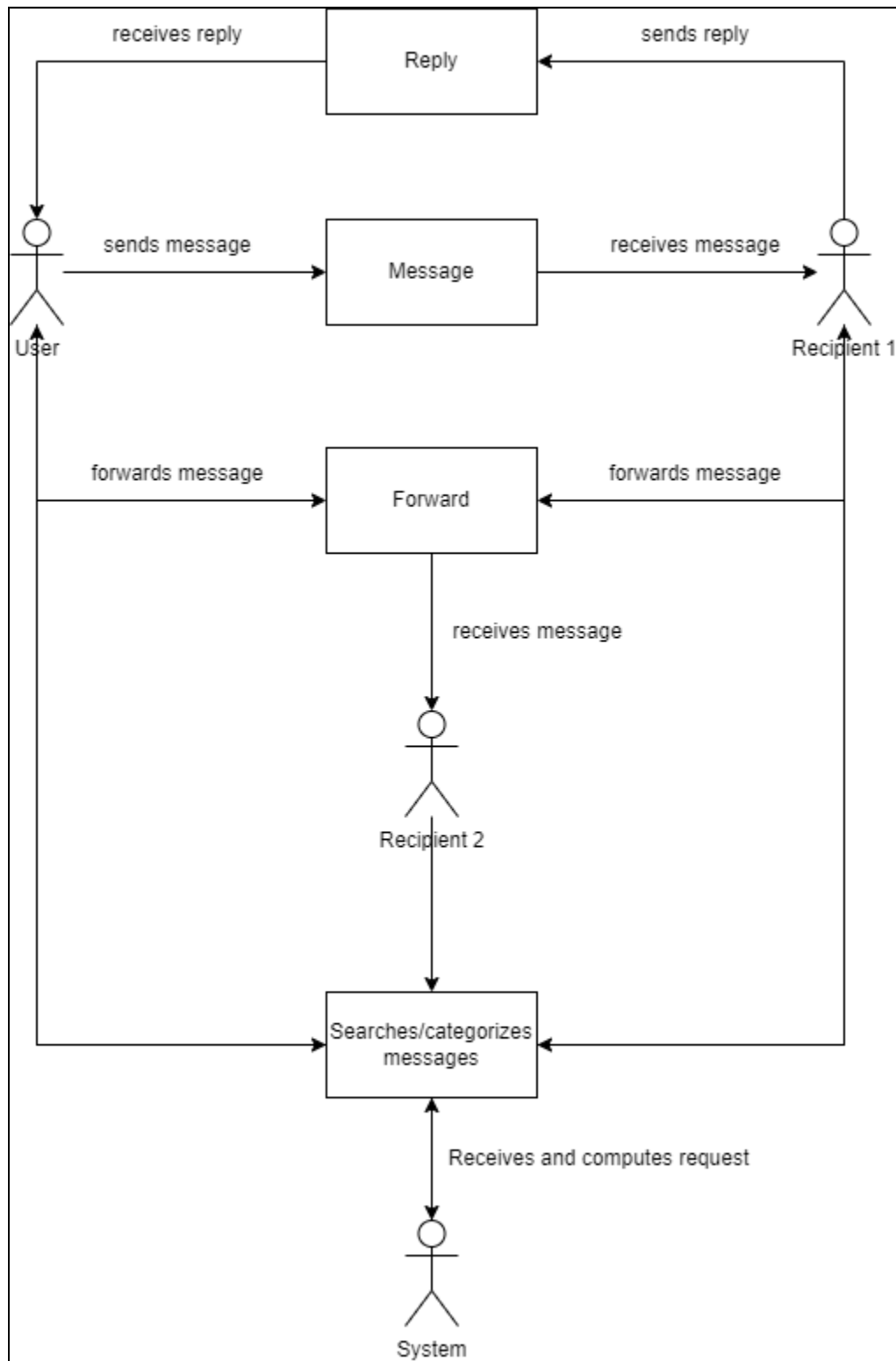
a. Functional requirements (5-7)

- i. View, edit, and delete messages, save a message under edition as draft
- ii. Send & receive messages
- iii. Reply & forward messages
- iv. Search messages by text query
- v. Send to multiple receivers (optional)
- vi. Scheduled message (optional)
- vii. Auto reply (optional)
- viii. Message organization

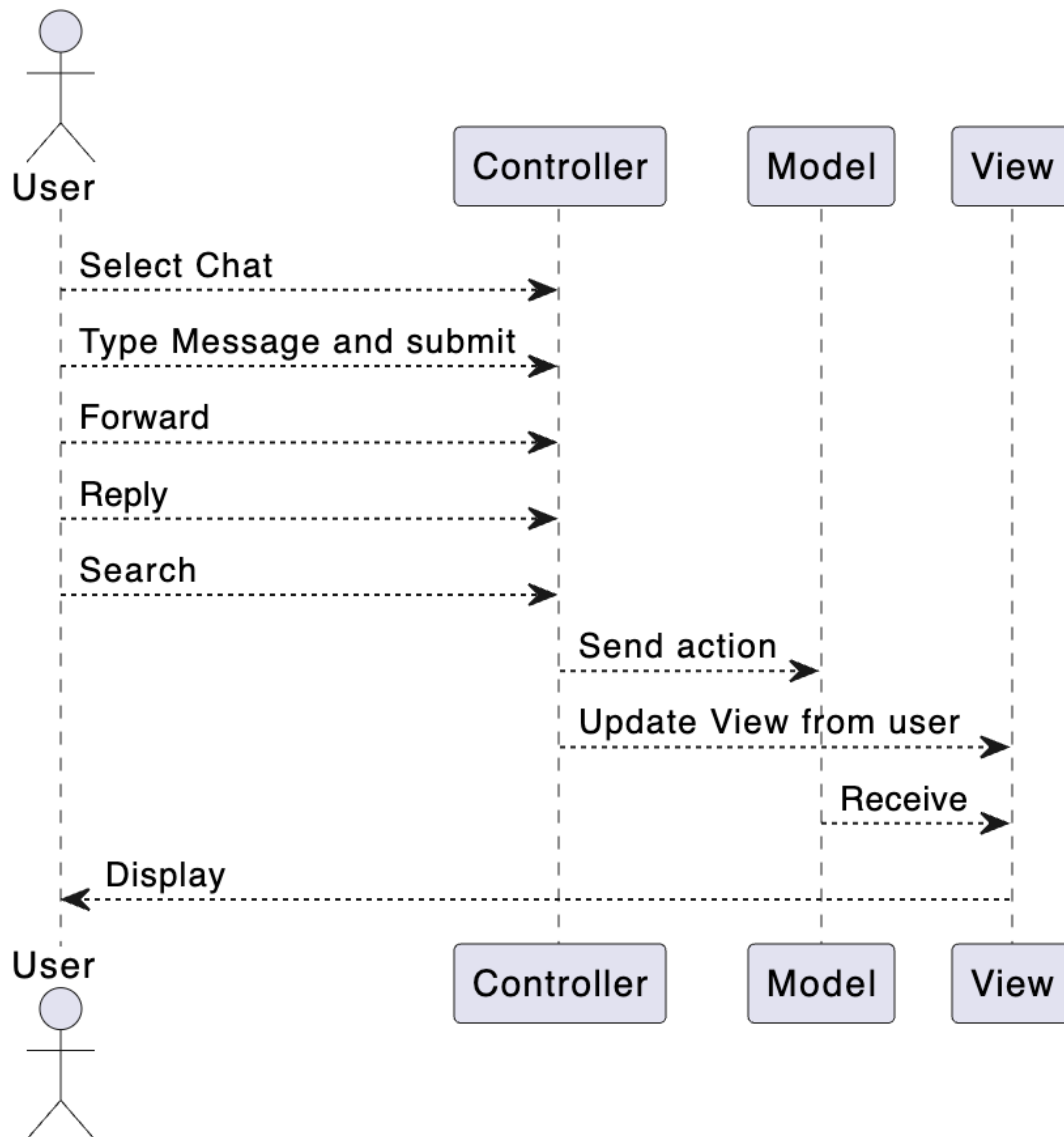
1. Categorize messages by phone number (contact name)
2. Conversation view: view all messages between you and a certain contact,
3. sorted by sending/receiving time
- ix. Zoom in/out and scroll whenever necessary
- x.
- b. Non-functional requirements (1 each for the categories) - Radhika
 - i. Product
 1. Usability - The software should have an intuitive user interface ensuring that users can easily manage messages without much training.
 2. Efficiency
 - a. Performance - should have low latency to ensure messages are received in real-time
 - b. Space - software should optimize storage space by compressing multimedia attachments
 3. Dependability - should have robust failover mechanisms and backup servers to ensure continuous service availability
 4. Security - messages and user data should be end-to-end encrypted, ensure that only the intended recipients can read the messages
 - ii. Organizational
 1. Environmental - software should be designed to minimize its carbon footprint by optimizing server infrastructure
 2. Operational - should have automated monitoring and alert systems in place to detect and resolve operational issues proactively
 3. Development - software should be modular and extensible, allowing for easy integration with third-party services and APIs, enabling developers to create custom features and extensions
 - iii. External
 1. Regulatory - The software must comply with relevant telecom regulations and data protection laws in all regions where it operates, ensuring legal compliance and user privacy.
 2. Ethical - software should have features to prevent the spread of hate speech, misinformation, and abusive content, promoting ethical communication among users.
 3. Legislative
 - a. Accounting - The software should have billing and usage tracking functionalities, allowing businesses to monitor their SMS usage, manage subscriptions, and view detailed billing reports.
 - b. Safety/security - The software should have a built-in emergency alert system, allowing users to send SOS

messages or alerts to designated contacts in case of emergencies, prioritizing safety and security.

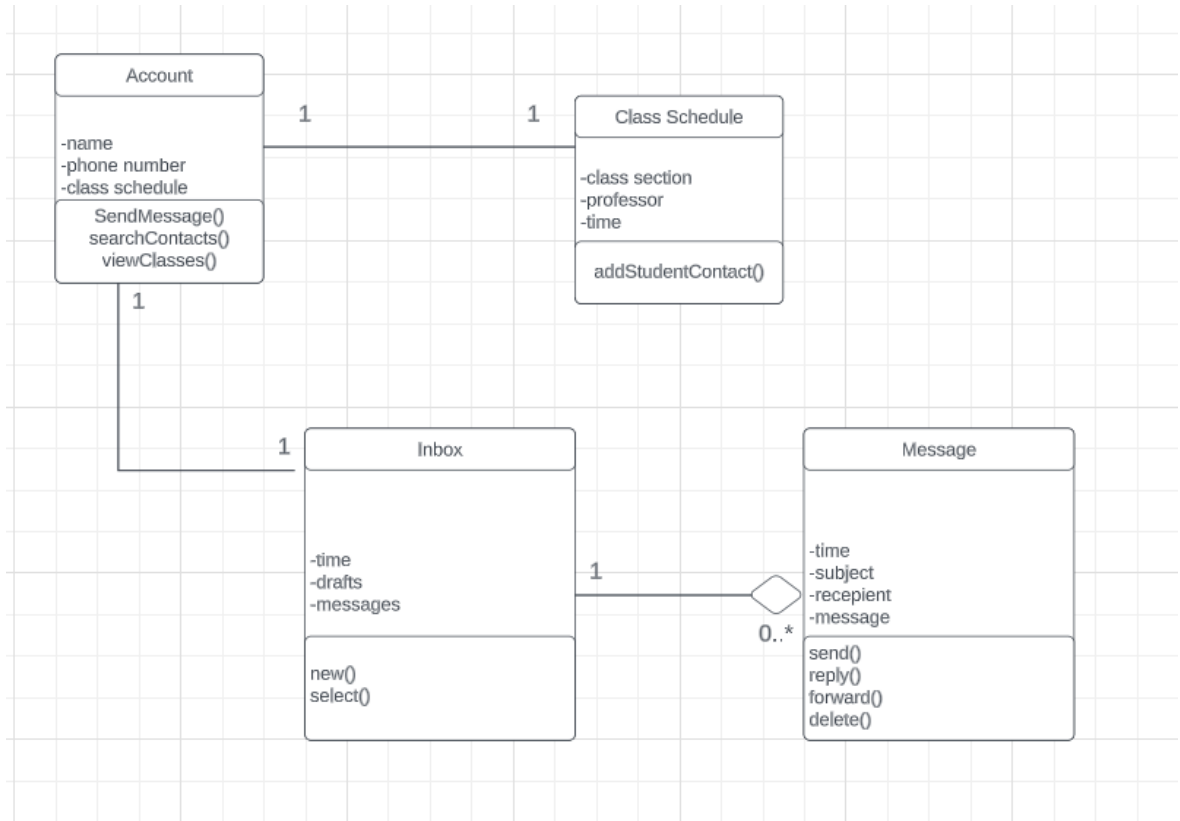
5. Use case diagram - Risheel



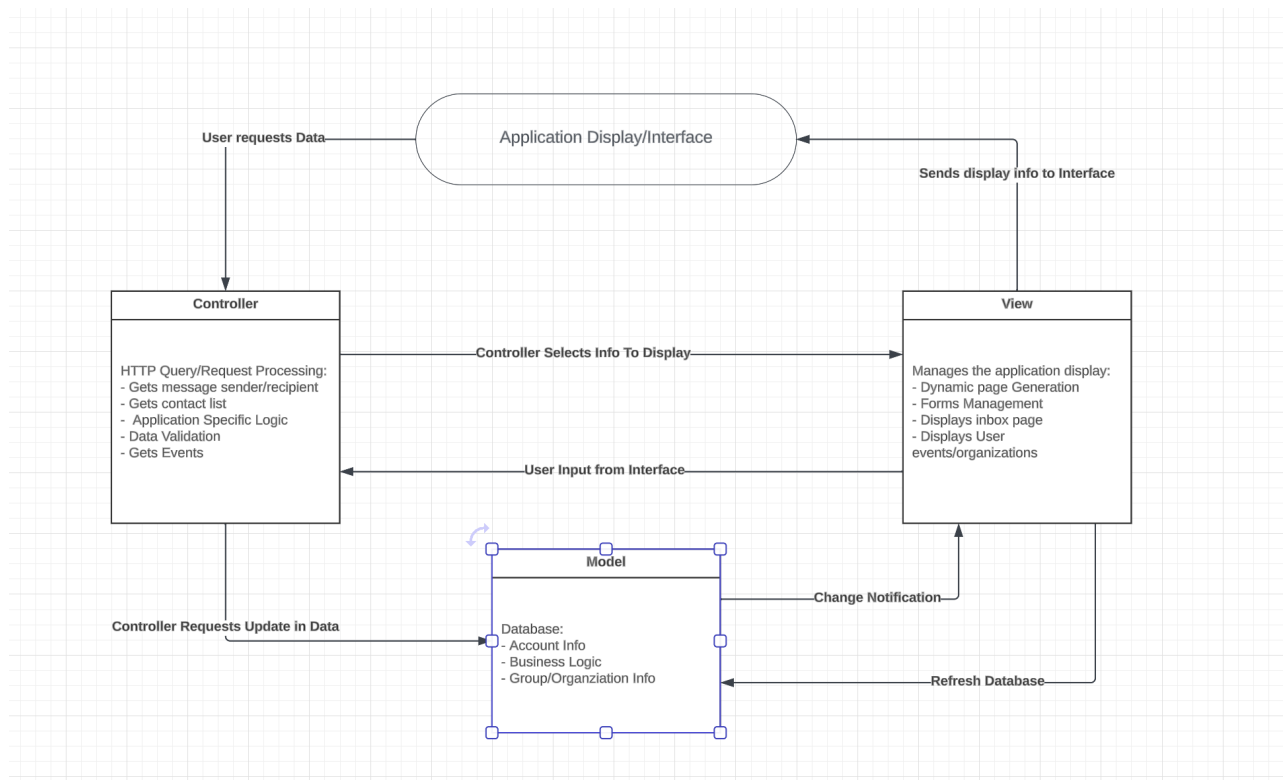
6. Sequence diagram for each use case - Aalhad Pathare



7. Class diagram - Sofia Trevino



9. Architectural pattern - MVC - need to make the design - Taylor



3. Project Scheduling, Cost, Effort and Pricing Estimation, Project duration and staffing:

3.1. Project Scheduling - Sofia

Start: February

End: June

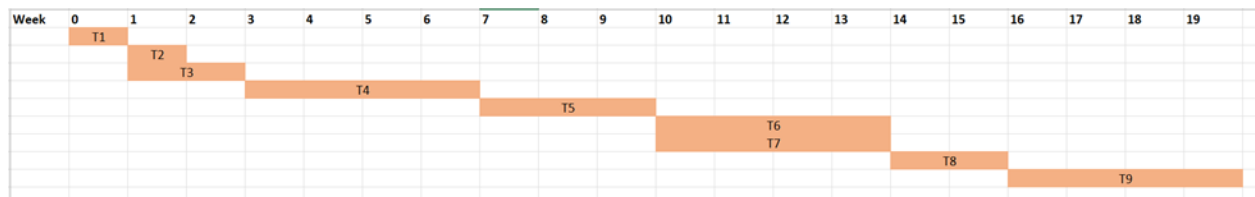
Total Duration: 5 months

- No weekends worked
- 40 hrs/wk

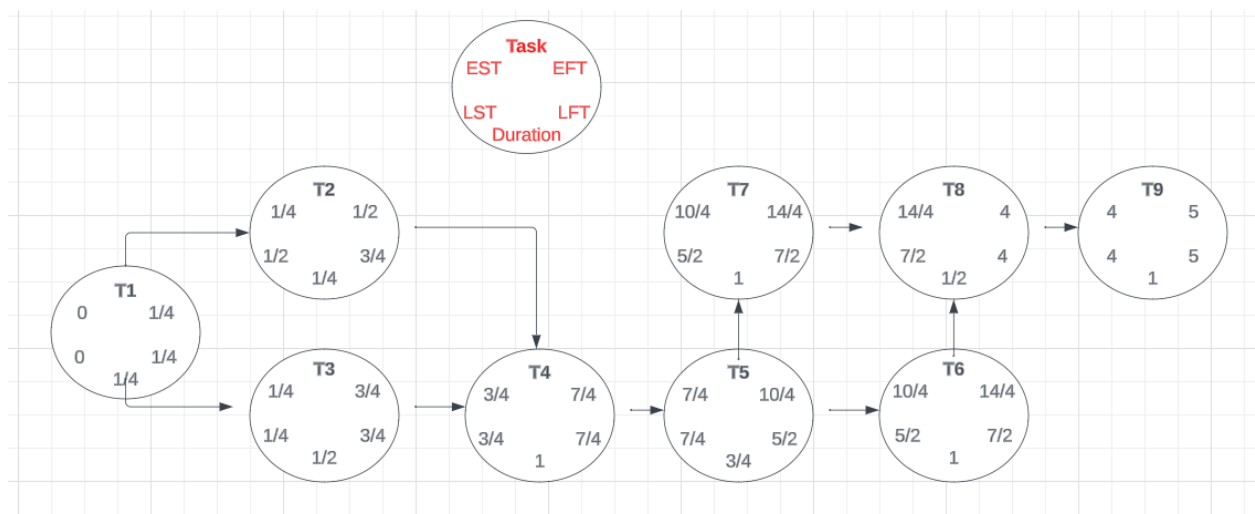
Task	Duration (weeks)	Dependencies
T1	1	
T2	1	T1
T3	2	T1
T4	4	T3
T5	3	T4
T6	4	T5
T7	4	T6
T8	2	T7
T9	4	T6, T8

Tasks

1. Gather requirements (1 week)
2. Create project plan (1 week)
3. Design user interface and database structure (2 weeks)
4. Develop core functionality (1 month)
5. Implement user interface ($\frac{3}{4}$ month ~ 3 weeks)
6. Perform test (1 month)
7. Perform debugging (1 month)
8. Collaborate with clients for feedback ($\frac{1}{2}$ month ~ 2 weeks)
9. Perform testing and quality assurance (1 month)



Activity on Node Representation:



Forward Pass on the Activity Node diagram shows that the earliest estimated finish time to complete all 9 tasks would be 5 months.

Backward Pass shows that the latest finish time to complete all 9 tasks would be 5 months.

Phase 1: Tasks 1-3 will take approximately 1 month as its just the initial stage of planning and designing the system and interface.

Phase 2: Tasks 4-8 will take approximately 3 months and is the longest because it involves creating the prototype. It includes writing the code for the core functionality, implementing the user interface, and running test and debugging based on client feedback on the prototype.

Phase 3: Task 9 will take approximately 1 month to complete as it performs the last set of tests and ensures it meets all standards for the users.

3.2. Cost, Effort, and Pricing Estimation - Taylor

Our group determined that the Function point method was best suited in estimating the effort, cost, and pricing for our project. Function point is a good user centric measurement and the core goal of our software is to gauge user interaction in order to ensure it provides what our customer needs. In addition, because of the nature of our project we feel that we are able to estimate, gauge the set of parameters including the number of user inputs, user queries, user outputs, data files, and external interfaces, all of which are accounted for using function point method.

	Function Category	Count	Complexity			Count * Complexity
			Simple	Average	Complex	
1	# of user input	8	3	4	6	24
2	# of user output	5	4	5	7	20
3	# of user queries	4	3	4	6	12
4	# of data files and relational tables	15	7	10	15	105
5	# of external interfaces	3	5	7	10	15

$$\text{GFP} = 24 + 20 + 12 + 105 + 15 = 176$$

- (1) Does the system require reliable backup and recovery? 3
- (2) Are data communications required? 5
- (3) Are there distributed processing functions? 4
- (4) Is performance critical? 3
- (5) Will the system run in an existing, heavily utilized operational environment? 2
- (6) Does the system require online data entry? 4
- (7) Does the online data entry require the input transaction to be built over multiple screens or operations? 5
- (8) Are the master files updated online? 4
- (9) Are the inputs, outputs, files, or inquiries complex? 2
- (10) Is the internal processing complex? 2
- (11) Is the code designed to be reusable? 4
- (12) Are conversion and installation included in the design? 5
- (13) Is the system designed for multiple installations in different organizations? 5
- (14) Is the application designed to facilitate change and ease of use by the user? 2

$$PCA = .65 + .01(3+5+4+3+2+4+5+4+2+2+4+5+5+2) = 1.15$$

$$FP = 176 * 1.15 = 202.4 \text{ FP}$$

Using 6 function points per person week we end up with: $202.4 / 6 = 34$ person weeks

Using a team size of 3: $34/3 = 11.333 = \text{(round up) 12 weeks}$ (this correlates to the number of weeks required to develop the product NOT the lifetime of the project) this number correlates closely with the duration of tasks 4 - 8 in the scheduling.

The total cost for the project will be the sum of the hardware, software, and personnel costs.

$$- 9500 + 3390 + 25000 + 465000 = 502,890.$$

We plan to make the software free to use as our audience is students who oftentimes do not have the financial resources to purchase software. Making our software free allows for easier access to its features. We plan on making revenue through the use of ad monetization and possibly subscription services.

3.3. Estimated cost of hardware products (such as servers, etc.) - Aalhad

When it comes to determining hardware costs for this application, it is important to consider a variety of factors, all that can contribute to this cost. It is important to consider the server infrastructure for any backend processes, the database to store messages, the Security measures to secure data, backup and redundancy of the data, and load balancing to handle a large number of requests.

If we were to build, deploy, and maintain our own server the cost would be substantial but maintain accessibility for our engineers to troubleshoot any issues. A server running an application of this scale with an initial ten terabytes of storage to redundantly store user information, videos, images, and files would cost close to **\$25000**.

Purchasing desktops or laptops to allow our 3 developers to create programs would be around \$4500. Any other office supply or physical hardware can be rounded up to an additional \$5000, totalling **\$9500** for the scope of the project.

3.4. Estimated cost of software products (such as licensed software, etc.) - Risheel

We need domain costs and domain protection costs for our SMS messenger software. Since these are in monthly plans, we would need to allocate around \$20 a month, or around \$240 a year.

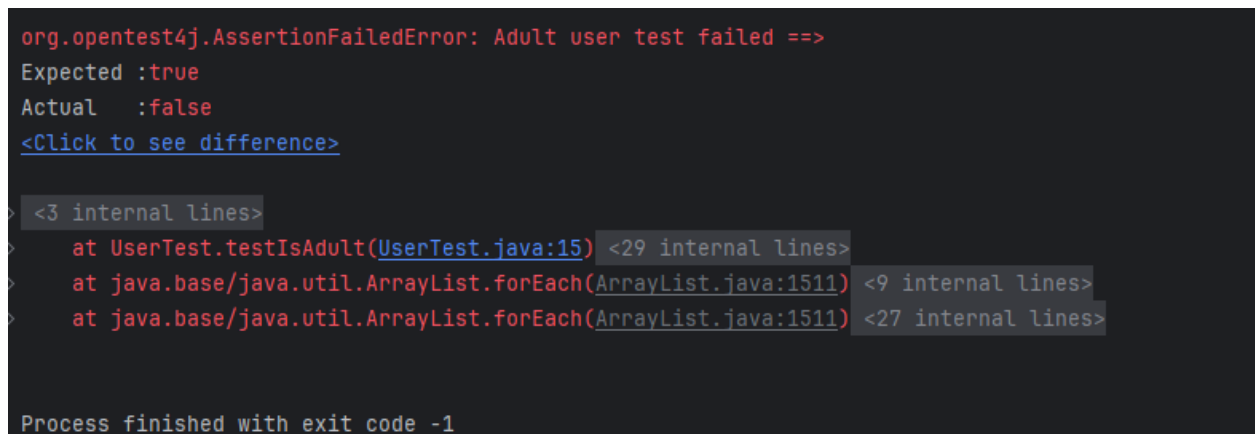
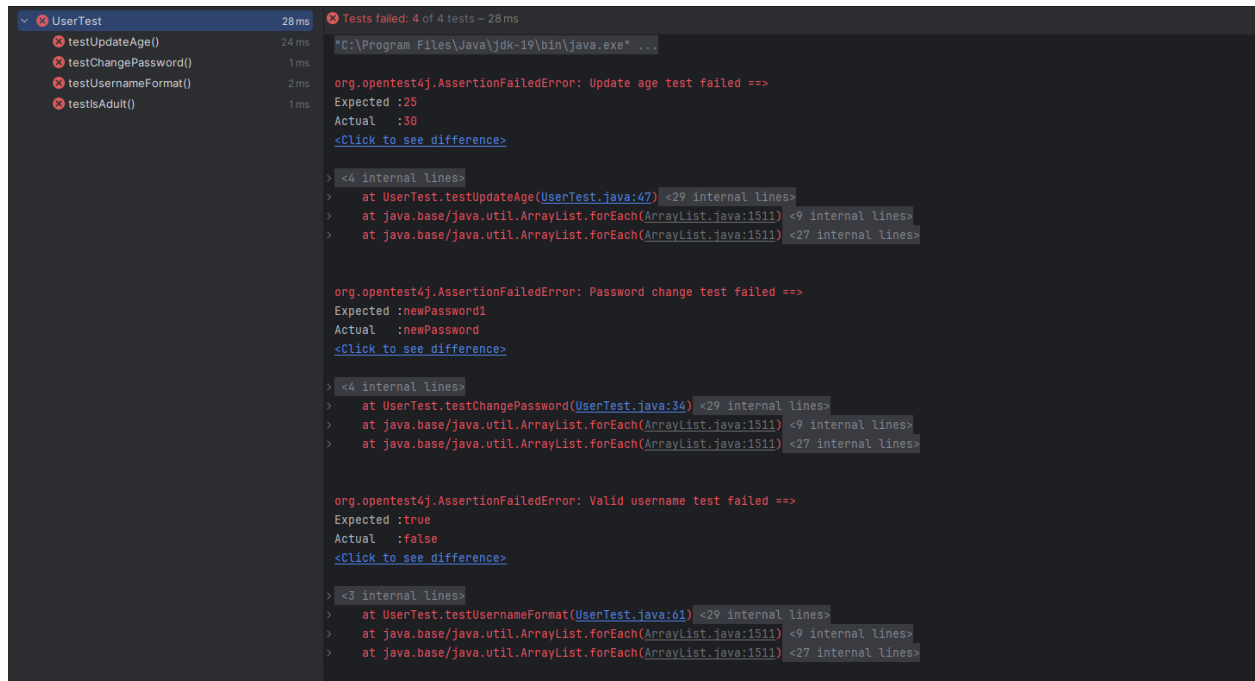
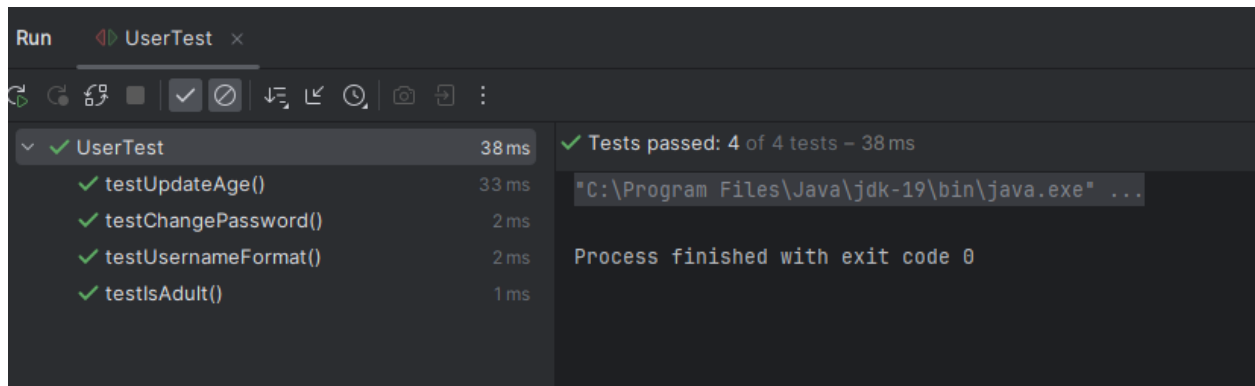
We also would need a software license to run our software, and Windows is a pretty good option as it costs up to \$3,000 per year. Twilio could also be used to test messages, and we would need to set aside around \$150 a year to send test text messages. These costs would add up to **\$3390** annualized

3.5. Estimated cost of personnel (number of people to code the end product, training cost after installation) - We will require a team of developers to create the project. In this specific case we are utilizing a team size of 3. Due to our smaller team size we have opted to invest more in the quality of our developers, which establishes an annual salary of \$155,000 for senior developers. So for our team it would require **\$465,000 per year** to pay all of the developers. Being a messaging software aimed for Student users we do not anticipate any costs for training as we aim for our product to be intuitive and straightforward to use.

4. Test plan for the software - Radhika, Tommy

For testing purposes, we implemented a sample unit for the User class containing the user's username, password and age. The class contains a constructor, getters, and setters for all data fields and some additional methods. Following are the methods included in the class, as well as the test plan to test the methods:

- a. `isAdult()` - method to check if the user is an adult
 - We created two users - one adult and one minor. We then tested the method to see if it showed the expected result for both kinds of users and also displayed the correct error message if an error was encountered.
- b. `changePassword(String newPassword)` - method to change the user's password
 - For testing this, we created a user with an old password and then changed the password to a new one. After updating it, we asked the test to see if the new password was returned when getting the user's password.
- c. `updateAge(int newAge)` - method to update the user's age
 - For testing this, we created a user with a set age. We then used the method to update the user's age and used `getAge()` to see the currently set age.
- d. `isValidUsername()` - method to check if the username format is valid
 - We created two users - one with a valid username and one with an invalid username. We then tested both users with the `isValidUsername()` method to see if it returned the expected results and corresponding error messages.



5. Comparison of our work with similar designs - Lay

Since our project is a messaging software, there are many established software that are similar to our design such as Discord, GroupMe, Messenger, Telegram, and Whatsapp. The main purpose of our project is for students to help them communicate with other students and meet other peers. The similarity between our design and others is to send direct messages to another person, reply messages, forward messages, and search messages. Our work is different from similar design by allowing students to meet up with peers, create and manage events, and be a platform for collaboration and communication between students.

6. Conclusion - Risheel

In conclusion, the SMS messaging software has a wide range of features, including sending texts, message management, and scheduled messages. This project prioritizes user experience, performance, and security. With this, we ensure a user-friendly interface and safe texting. Externally, the software follows regulatory compliance, ethical communication, and user safety through accounting and security features. Overall, the SMS messaging software provides a secure, efficient, and user-friendly messaging platform with a strong emphasis on functionality, dependability, and ethical use.

7. References (IEEE citation)

- [1] BuiltIn, "Senior Software Engineer salary in US," 2023 Senior Software Engineer Salary in US | Built In, <https://builtin.com/salaries/dev-engineer/senior-software-engineer> (accessed Nov. 14, 2023).
- [2] Twilio, "Twilio", <https://www.twilio.com/> (Accessed Nov. 14, 2023)
- [3] Microsoft, "Windows License Pricing", <https://www.microsoft.com/> (Accessed Nov. 14, 2023)
- [4] AWS, "AWS Pricing Calculator", <https://calculator.aws/> (Accessed Nov. 14, 2023)
- [5] Admin, "How much office space do I need? (calculator & per person standards)," AQUILA Commercial, (accessed Nov. 15, 2023).
- [6] C. Probst, "Best laptops for programming in 2023," TechRadar,
- [7] Discord, "Your Place to Talk and Hang Out," <https://discord.com/> (Accessed Nov. 16, 2023)
- [8] GroupMe, "Where your group chats best," <https://www.groupme.com/en-US/> (Accessed Nov. 16, 2023)
- [9] Messenger, "Messenger," <https://www.messenger.com/> (Accessed Nov. 16, 2023)
- [10] Telegram, "A New Era of Messaging," <https://telegram.org/> (Accessed Nov. 16, 2023)
- [11] Whatsapp, "WhatsApp: Secure and Reliable Free Private Messaging and Calling," <https://www.whatsapp.com/> (Accessed Nov. 16, 2023)

8. Presentation Slides

9. To post to GitHub

- **Deliverable 2 report**
- **Unit test code for sample unit**

- **Implementation code (if any)**
- **Presentation slides**