**CS3354 Software Engineering**

**Final Project Deliverable 2**

Rilakkuma

Taylor Benavides

Kayla Carroll

Luis Flores

Jorge Martinez

Brent Powell

Kairo Sauceda

Renno Sullivan

1)

# DELEGATION OF TASKS

## 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. PEOPLE: Kayla

## 2. [5 POINTS]

Everything required and already submitted in Final Project Deliverable. Please specify this part as “Project Deliverable 1 content”.

PEOPLE: Brent

## 3. [50 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: duration and staffing:

Include a detailed study of project scheduling, cost and pricing

### 3.1. [20 POINTS]

Project Scheduling. Please note that what you present should be the timeline of the project designed, NOT the time you’ve spent on it. Use an automated tool (such as MS Project) to plan a schedule for your project. It should include tasks, durations, and dependencies for your project provided on a table (similar to Figure 23.5), as well as an activity bar chart (similar to Figure 23.6) drawn using an automated tool (such as MS Project). A guideline document on how to install MS Project from Microsoft Imagine link, as well as a sample MS Project file for helping you prepare a timeline graph were already provided as part of hw7 and provided again attached to this document. Also, remember that MS Project is installed and ready for use in UTD Open Lab computers.

PEOPLE: Taylor

### 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. Some cost modeling techniques you may use are listed as follows:

1. *Function Point*

*Or any of the following COCOMO II estimation models*

1. *Application composition*
2. *Early design*
3. *Post-architecture*

PEOPLE: Renno, Kayla

i. Function Point

### 3.3. [5 POINTS]

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

PEOPLE: Brent

### 3.4 [5 POINTS]

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

PEOPLE: Jorge

### 3.5 [5 POINTS]

Estimated cost of personnel (number of people to code the end product, training cost after installation)

PEOPLE: Taylor

## 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) PEOPLE: Luis

## 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.

PEOPLE: Kairo, Renno

## 6. [10 POINTS]

Everything required and already submitted in Final Project Deliverable. Please specify this part as “Project Deliverable 1 content”.

PEOPLE: Brent, Luis, Kayla

1. Write: Brent, Luis
2. Review: Kayla

## 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://ieeedataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guideli nes.pdf](https://ieeedataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf)

It means that your references should be numbered, and these numbers properly cited in your project report.

PEOPLE: Brent, Taylor, Luis

## 8. [5 POINTS]

Your presentation slides. No min/max number of slides enforced. Please make sure that you can complete presentation within 12 (twelve) 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

PEOPLE: Taylor, Brent

* Cost estimation

PEOPLE: Renno, Kayla

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

PEOPLE: Taylor

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

PEOPLE: Taylor, Kayla

* Use case diagram

PEOPLE: Renno, Jorge

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

PEOPLE: Brent, Kairo

* Class diagram

PEOPLE: Taylor, Brent, Kayla

* Architectural design

PEOPLE: Taylor, Luis

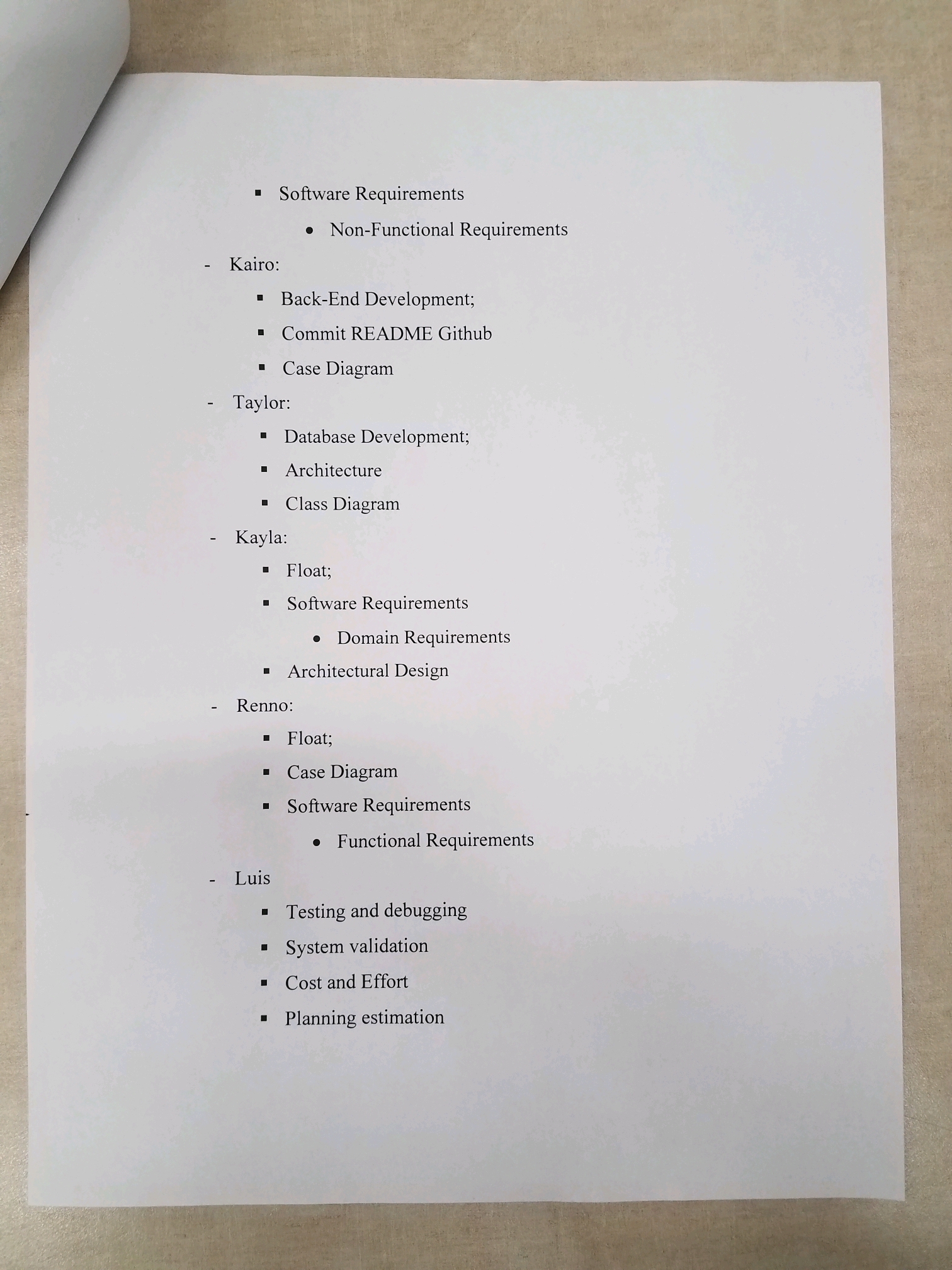
* Model-View-Controller (MVC) pattern (similar to Figure 6.6)
* Layered architecture pattern (similar to Figure 6.9) 4
* Repository architecture pattern (similar to Figure 6.11)
* Client-server architecture pattern (similar to Figure 6.13)
* Pipe and filter architecture pattern (similar to Figure 6.15)
* 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

PEOPLE: Kayla

**\*\*\*BEGINNING OF DELIVERABLE ONE\*\*\***

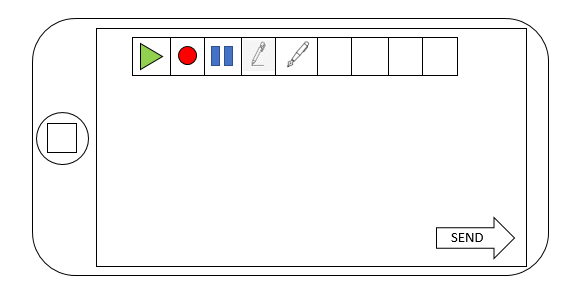
**2. Everything required and already submitted in Final Project Deliverable**

1



Trying to communicate homework help across text messages is difficult. Explanations oftentimes need the explainer to talk along with what they are writing in order to correctly convey their ideas. Rilakkuma is a web-based application that focuses on sharing a real-time whiteboard between users that can text and talk about it with each other. The users can login using an account and password they provide and connect with other users.

Example:



1.1 COMPLETE

1.2 COMPLETE

1.3 COMPLETE

1.4 COMPLETE

1.5 COMPLETE

1.6 COMPLETE

Team project repository URL: <https://github.com/tscott32/3354-TheMorningParty.git>

2. - Taylor: *Delegation of Tasks*

3. - Jorge: *Incremental Software Process Model*

4. - Taylor/Kayla: *Software Requirements*

5. - Renno/Jorge: *Use Case Diagram*

6. - Brent/Kairo: *Sequence Diagrams*

7. - Kayla/Taylor/Brent: *Class Diagrams*

8. - Taylor/Luis: *Architectural Design -* *MVC*

3. [5 POINTS] Which software process model is employed in the project and why.

(Ch 2)

The software process model employed in our project is the incremental development software process model because it allows us to make changes and correct errors incrementally with ease if the latest functionality increment does not meet the project requirements.

4. [15 POINTS] Software Requirements including

4.a.) [5 POINTS] Functional requirements. To simplify your design, please keep your functional requirements in the range minimum 5 (five) to maximum 7 (seven). (Ch 4)

* User must access service i.e., website/mobile application
* User must be able to login/logout
* Search/add/remove contacts
* Ability to send text messages and join calls between mutually added contacts
* Ability to create/edit/delete/share/print “whiteboards”
* Ability to add and edit content to whiteboards

4.b.) [10 POINTS] Non-functional requirements (use all non-functional requirement types listed in Figure 4.3 - Ch 4)

* Product Requirements
  + Usability Requirements

1. Easy and/or simple to use UI
2. Easy to learn application
   * Efficiency Requirements
     + Performance Requirements
3. Minimal Latency (< 12ms)
   * + Space Requirements
4. Text messages < 4MB, whiteboard < 12MB
   * Dependability Requirements
5. Failure Rate < 4%
   * Security Requirements
6. Password specifications
7. Reset password authentication

* Organizational Requirements
  + Environmental Requirements

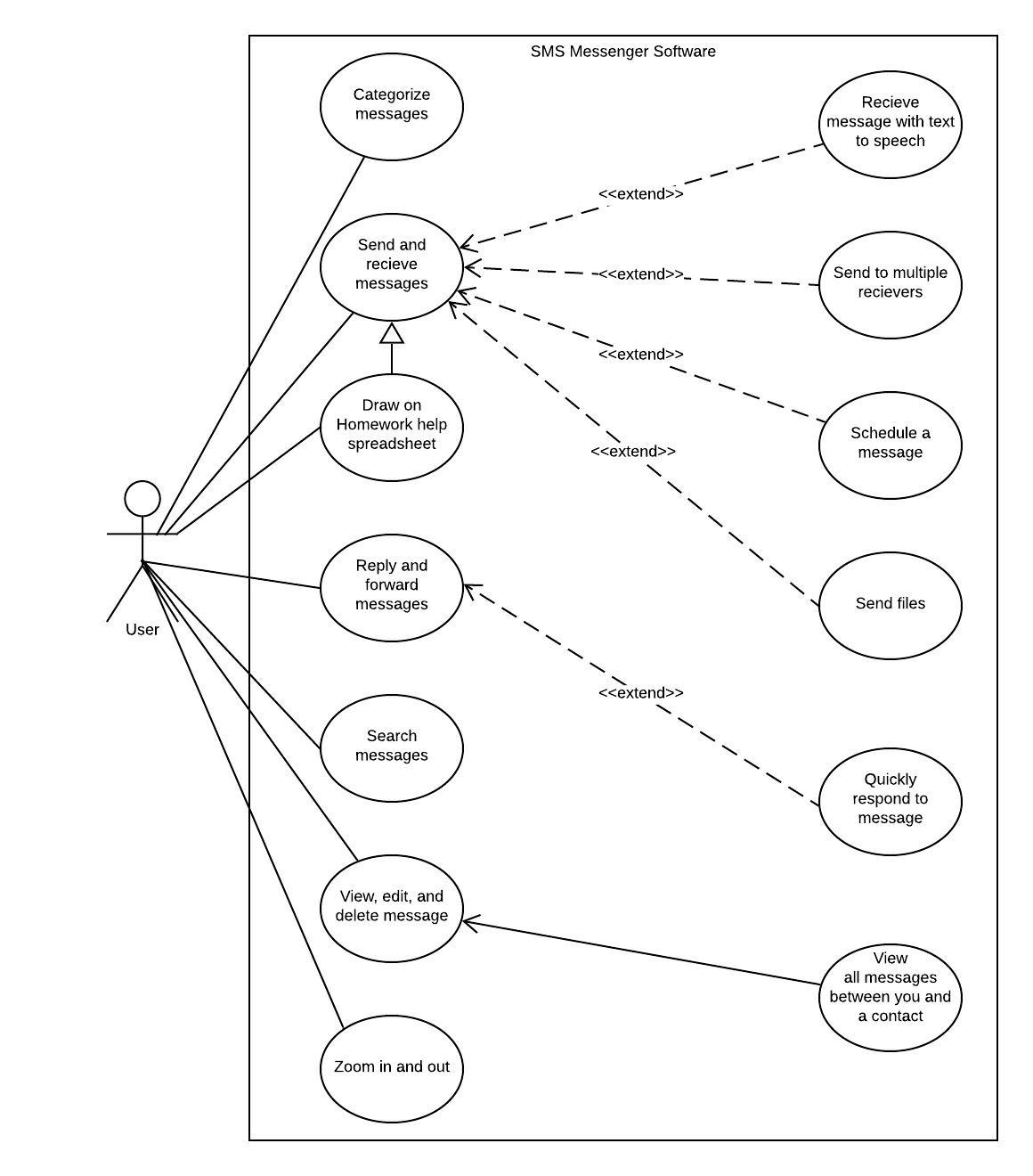
1. User may be in noisy environment: built-in background noise suppression
   * Operational Requirements
2. Requires space on user’s PC to download application
3. Requires stable wifi
4. Minimum 30kbps
   * Development Requirements
5. Html, CSS, PHP, SQL, javaScript

* External Requirements
  + Regulatory Requirements

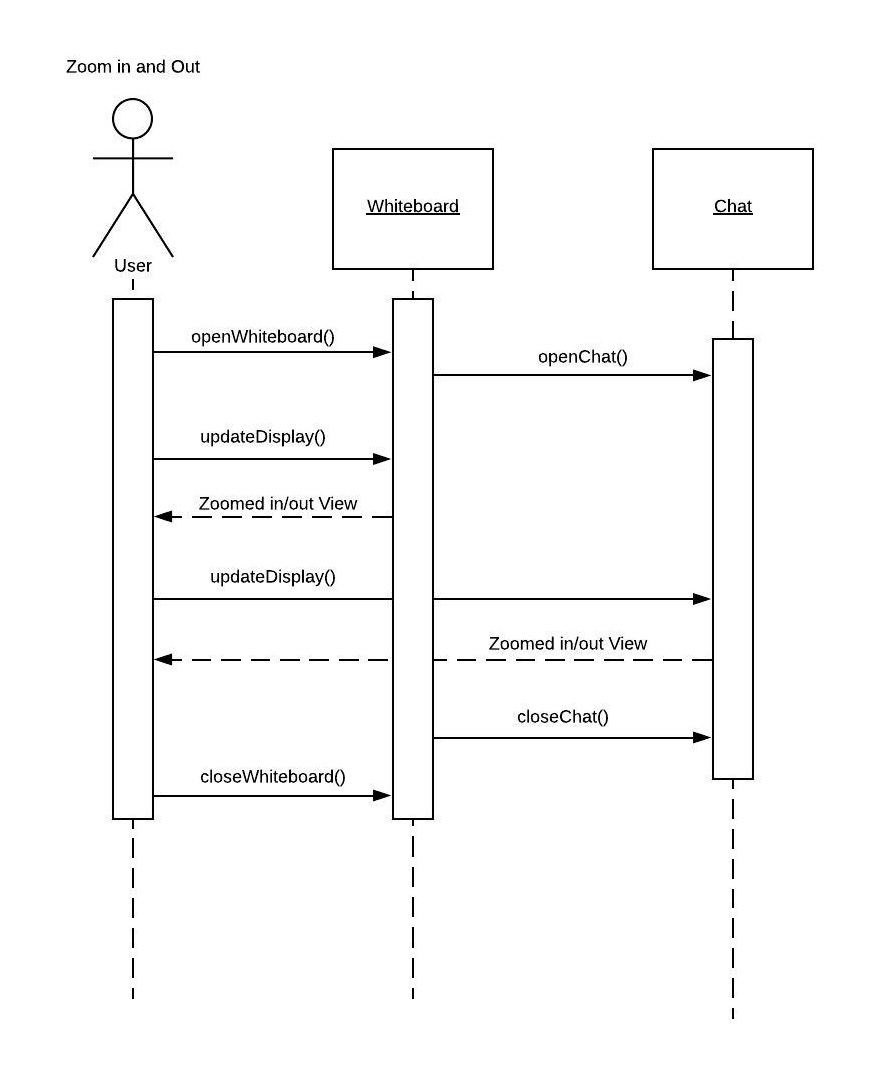
1. Function in accordance with TCPA
2. Users agreeing to Terms of Service and Privacy Policy
   * Ethical Requirements
3. Assure users their personal information (address, DOB, any payment methods, etc.) will never be shared
   * Legislative Requirements
     + Accounting Requirements
4. Limit the Use of Storage of Sensitive Data
   * + Safety/Security Requirements
5. Web Application Security Standards

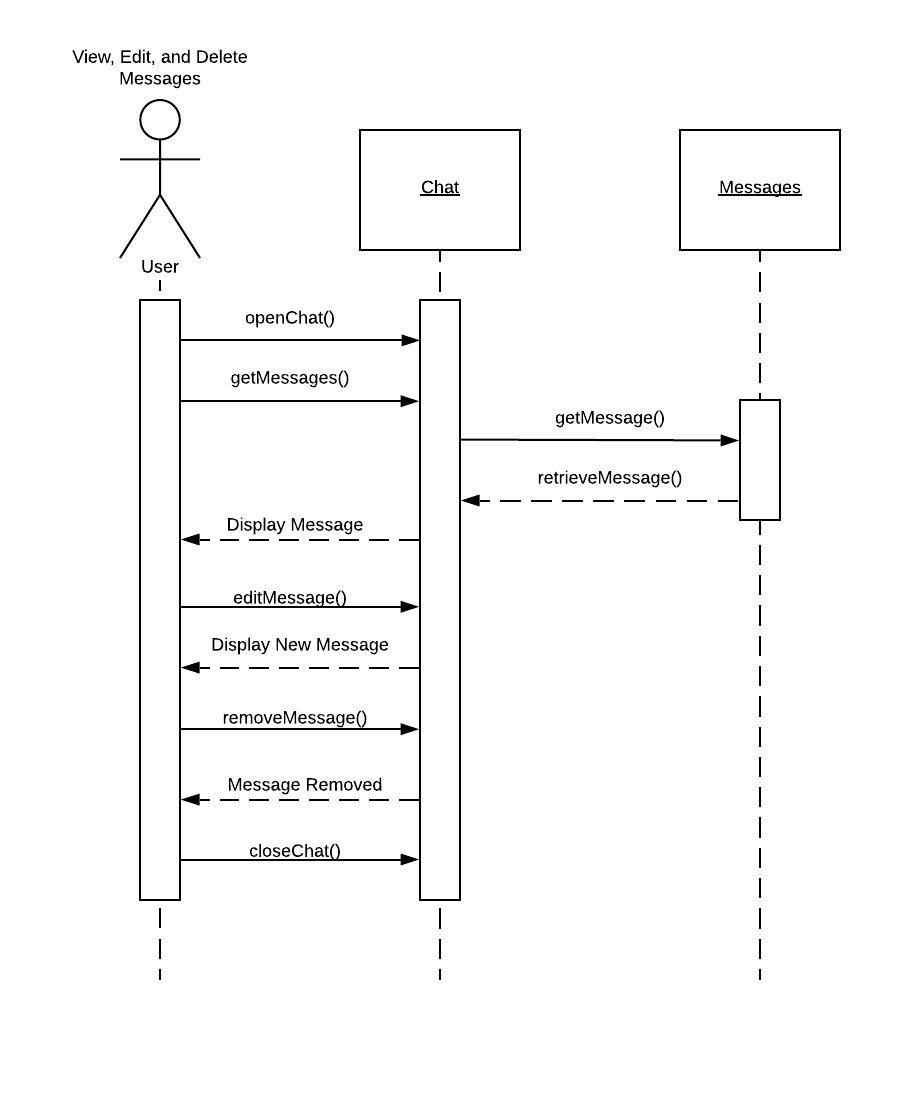
[https://hhs.texas.gov/sites/default/files/documents/doing-business-with-hhs/contracting/web-mobile-minimum-security-standards.pd](https://hhs.texas.gov/sites/default/files/documents/doing-business-with-hhs/contracting/web-mobile-minimum-security-standards.pdf)

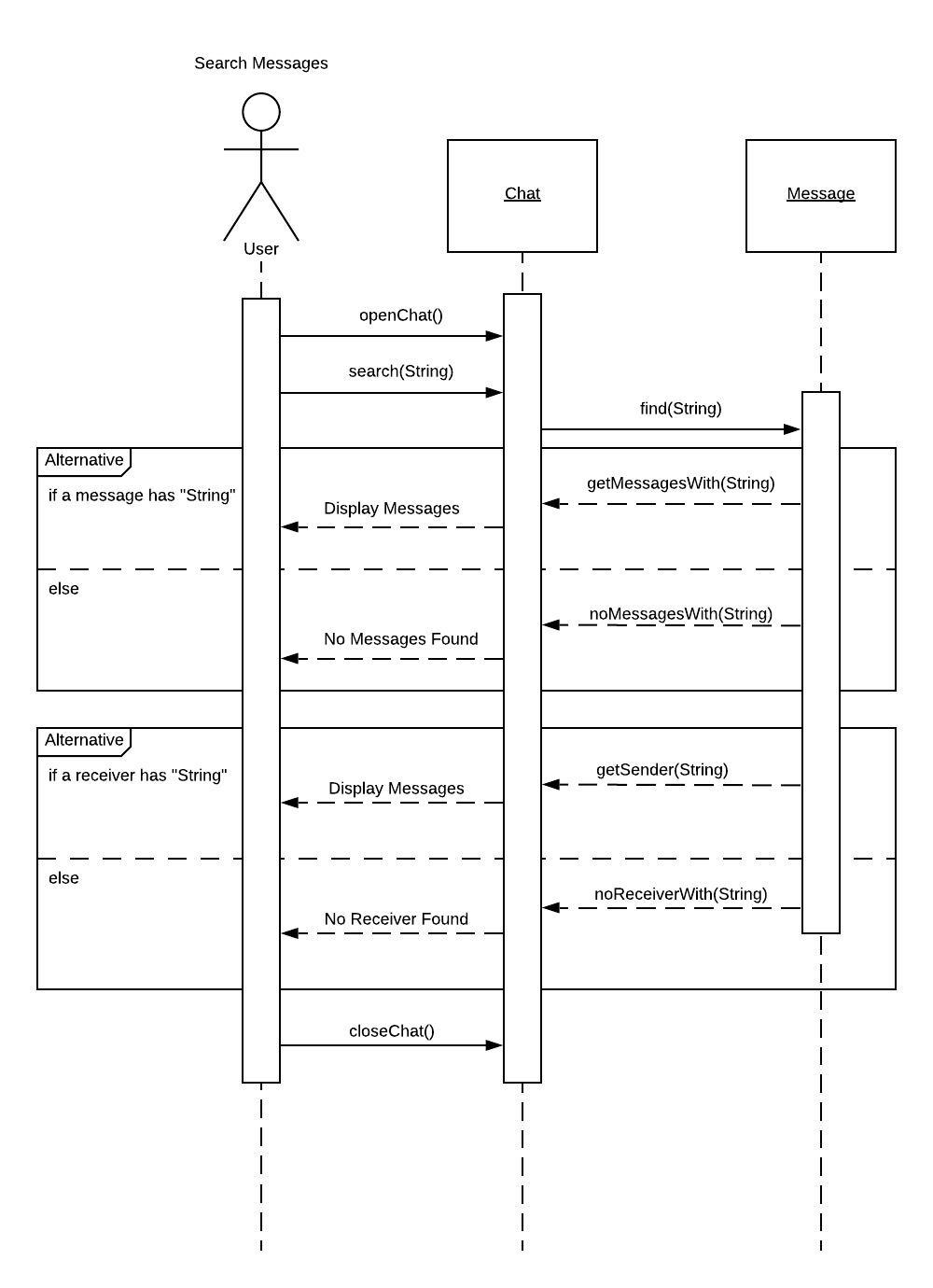
5.

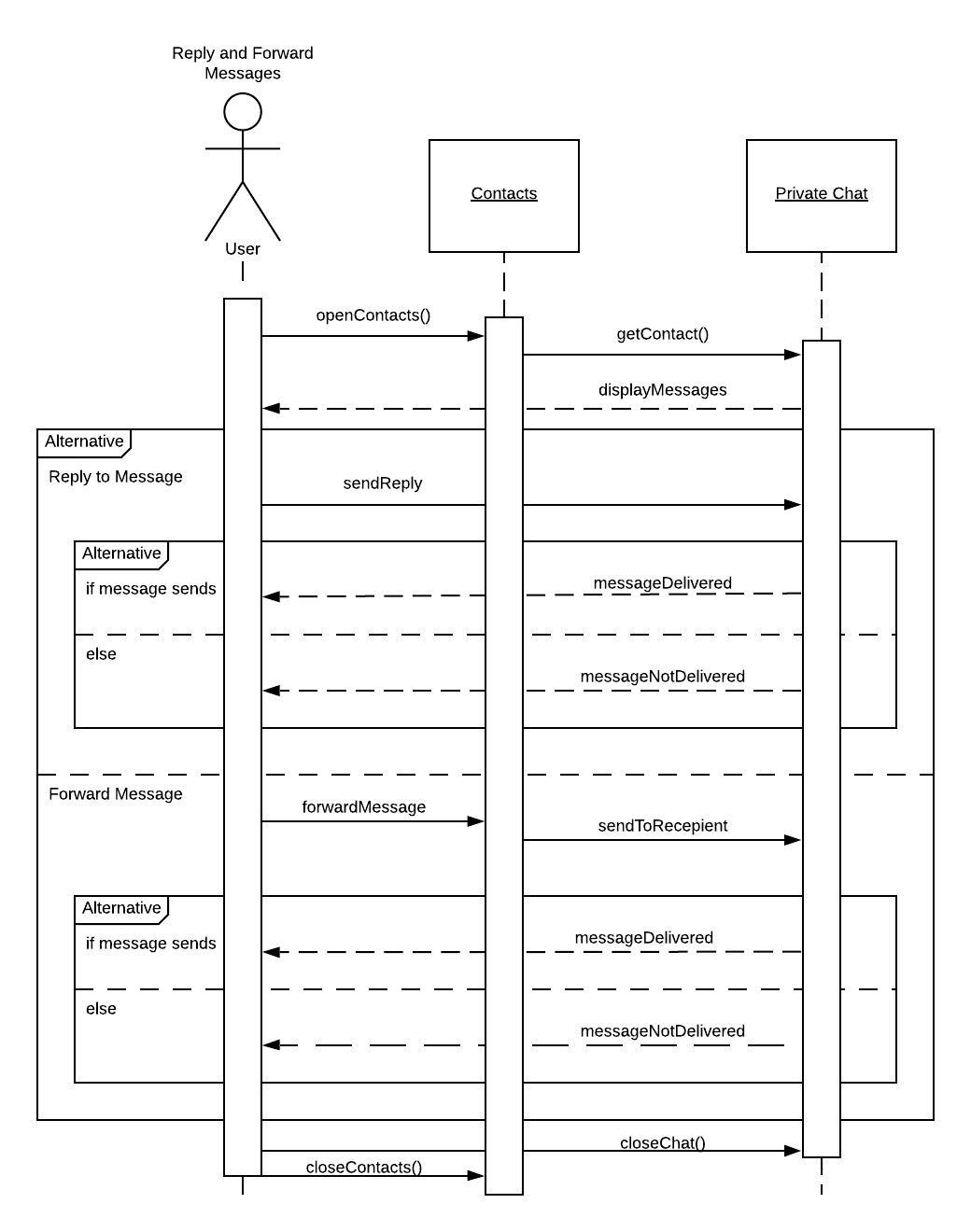


6.

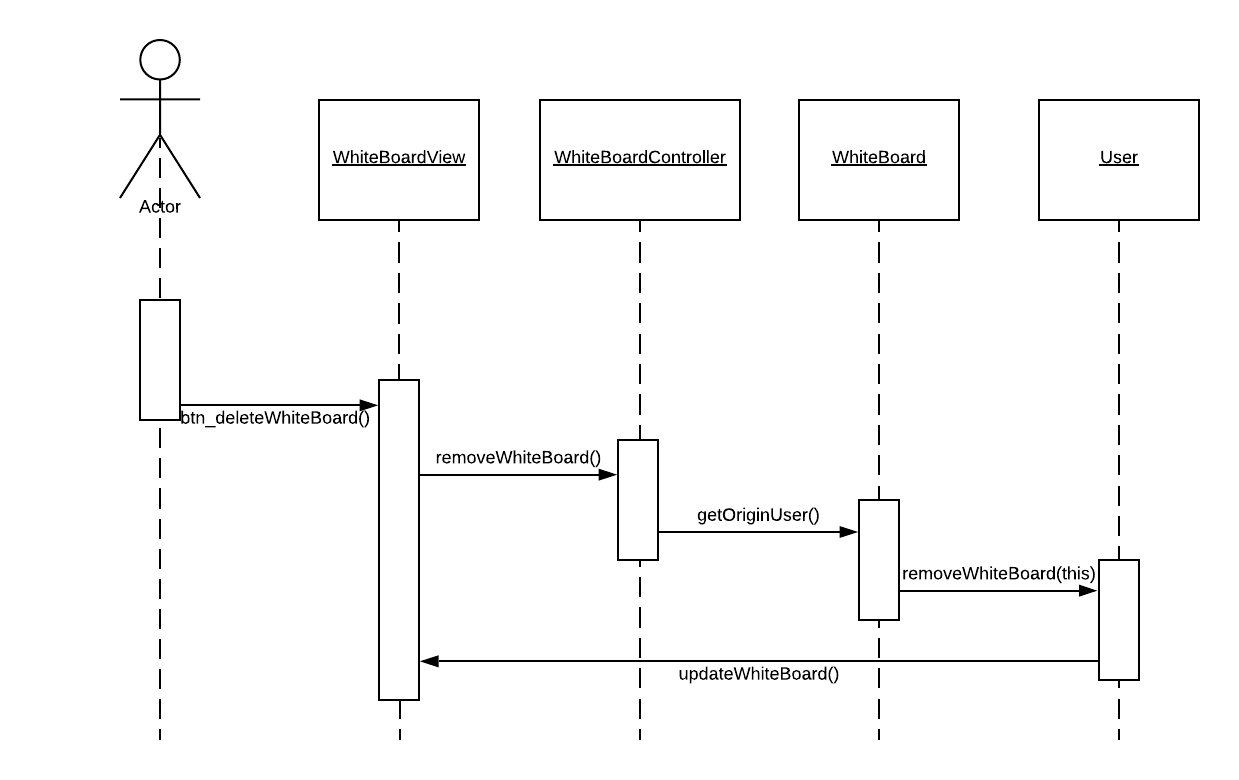




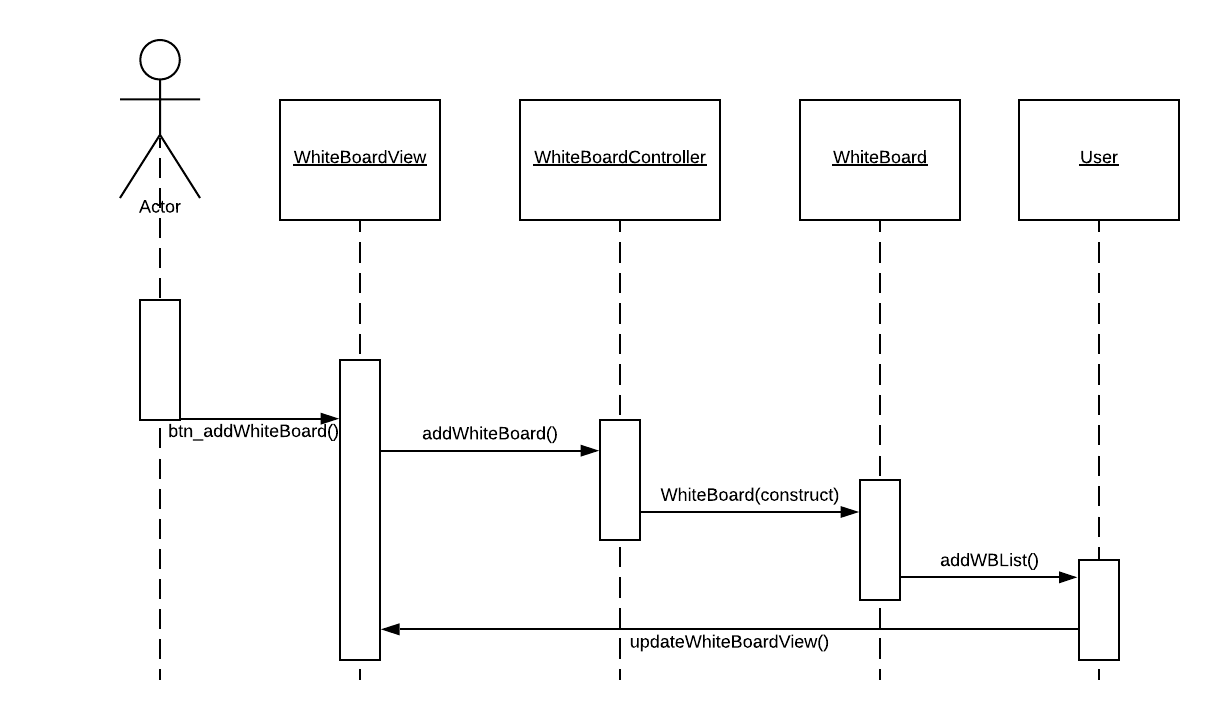




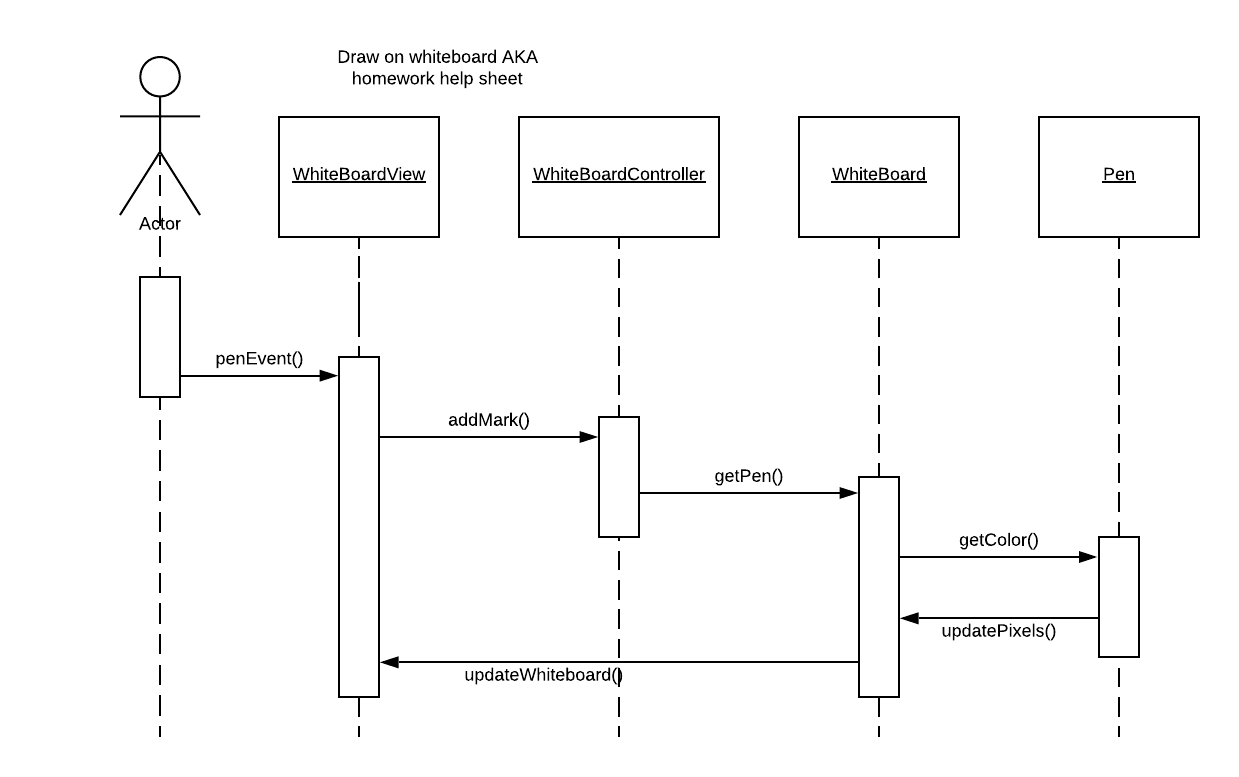
Delete Whiteboard



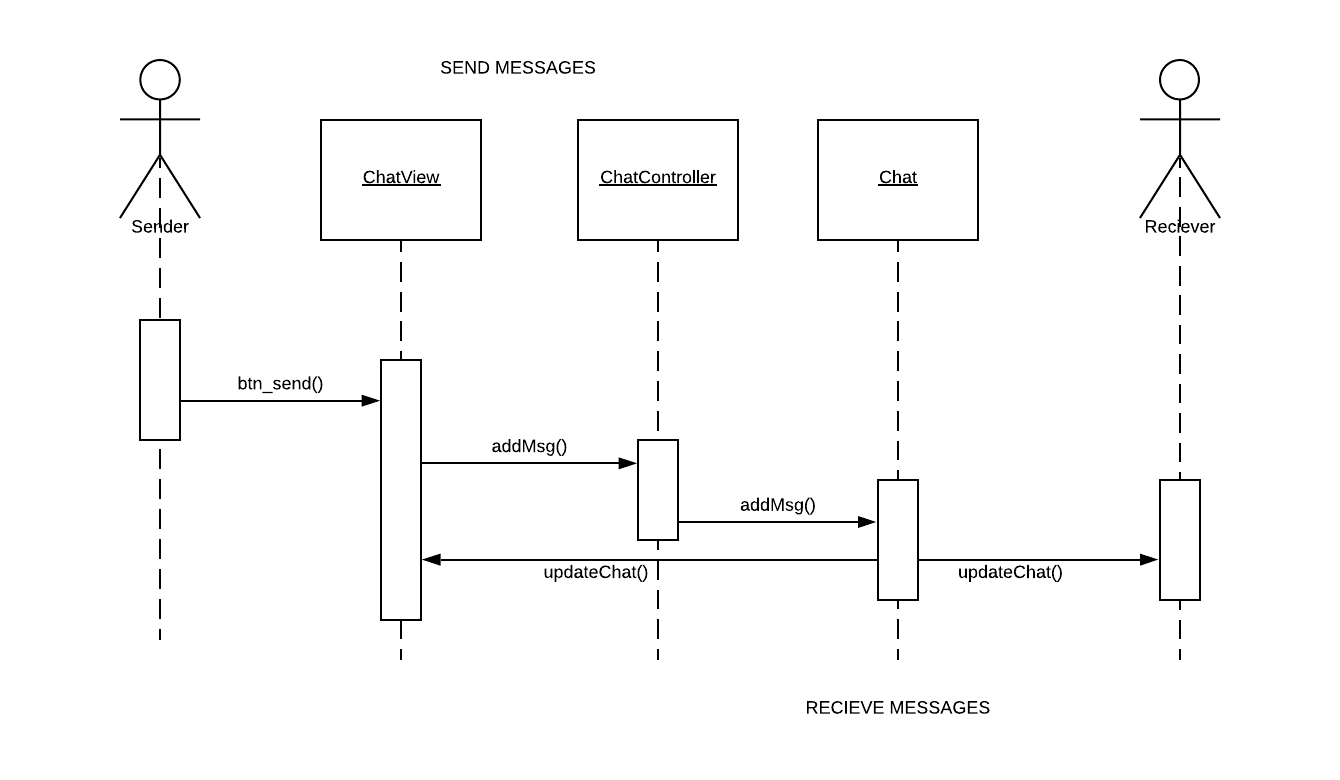
Create Whiteboard



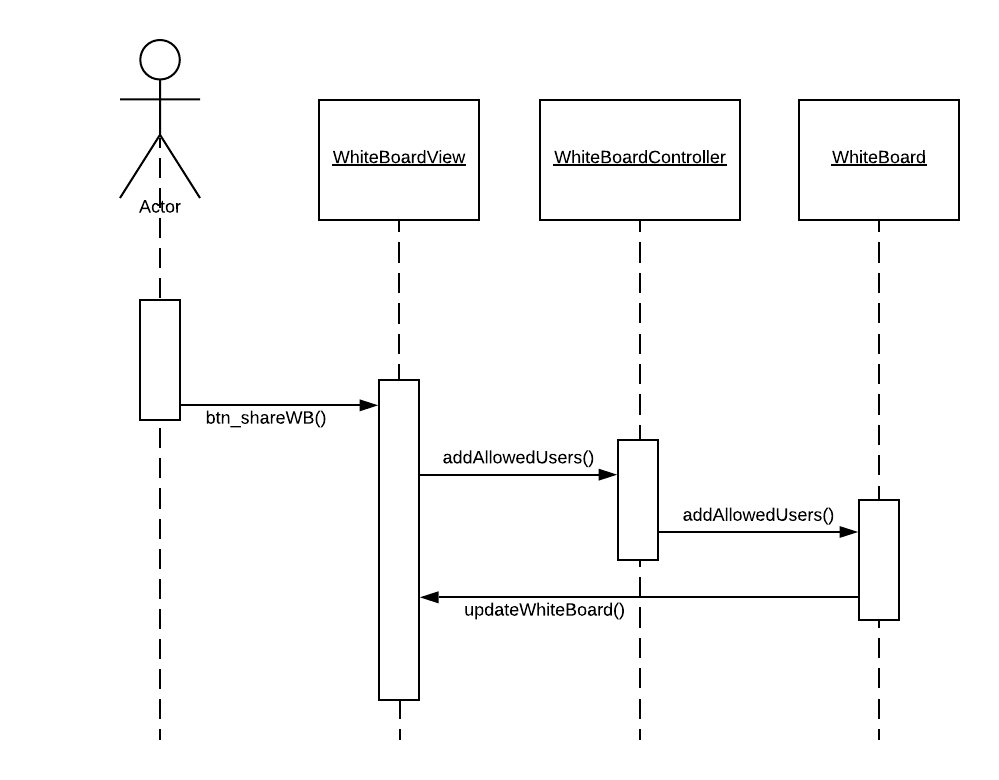
Edit whiteboard



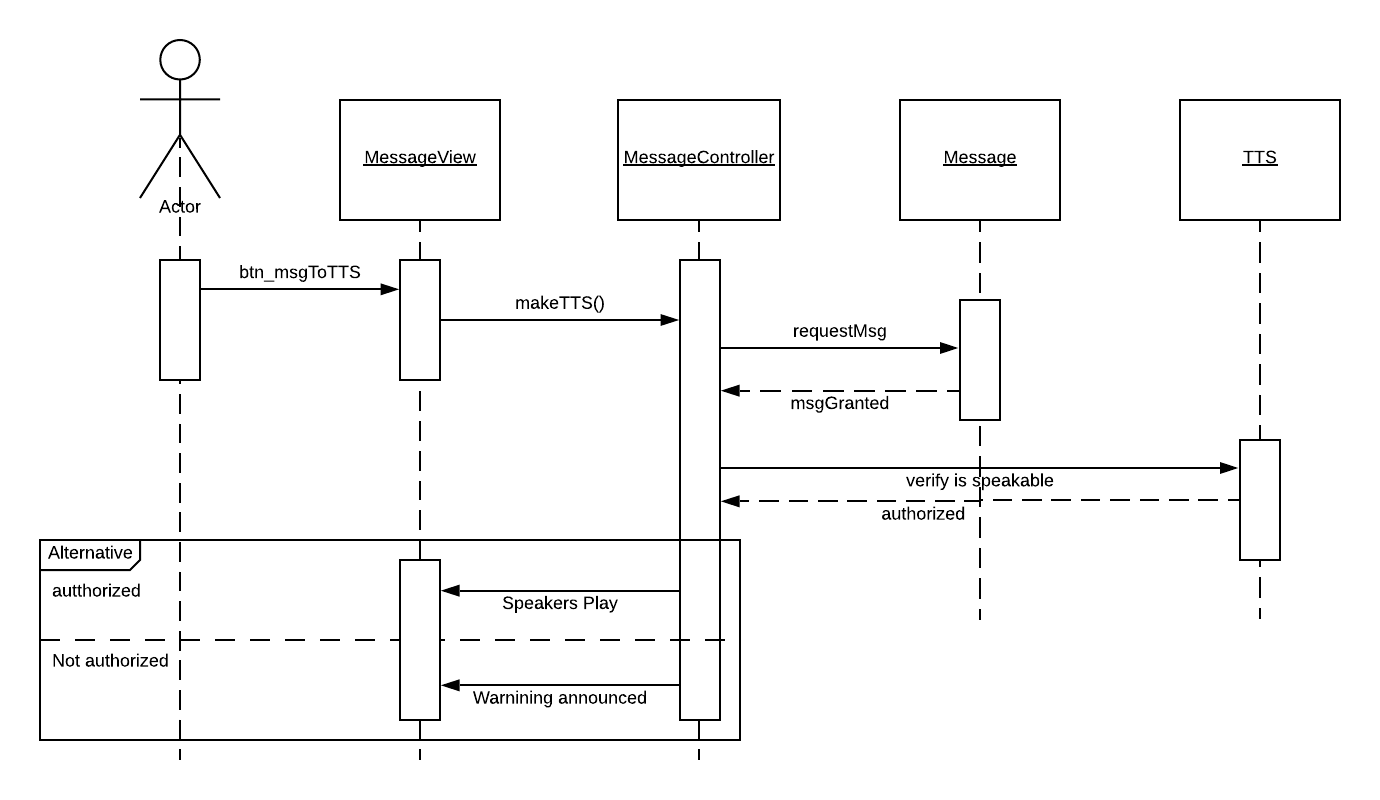
Send and Receive messages



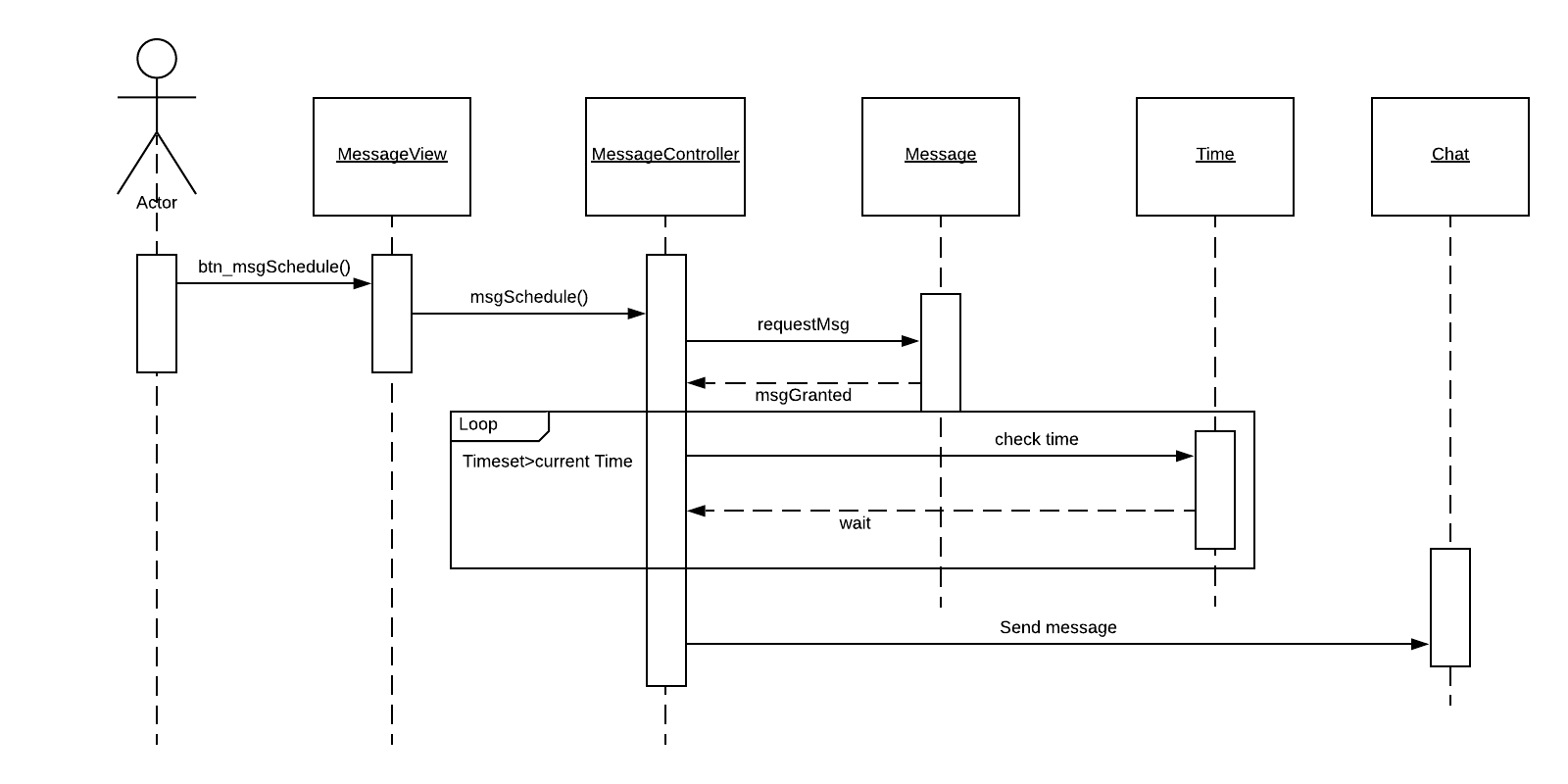
Share Whiteboard



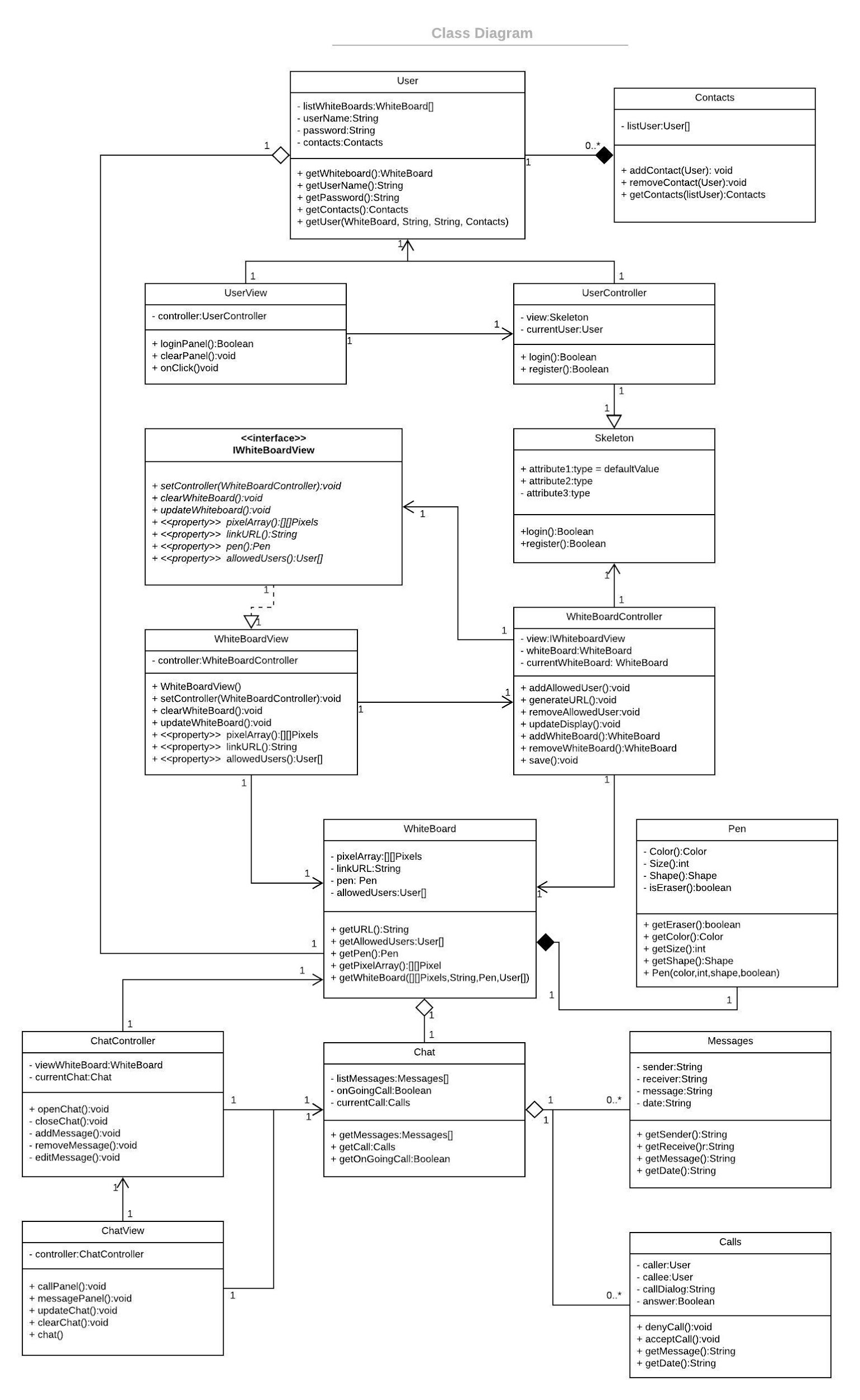
Receive message with text to speech



Schedule message



7.

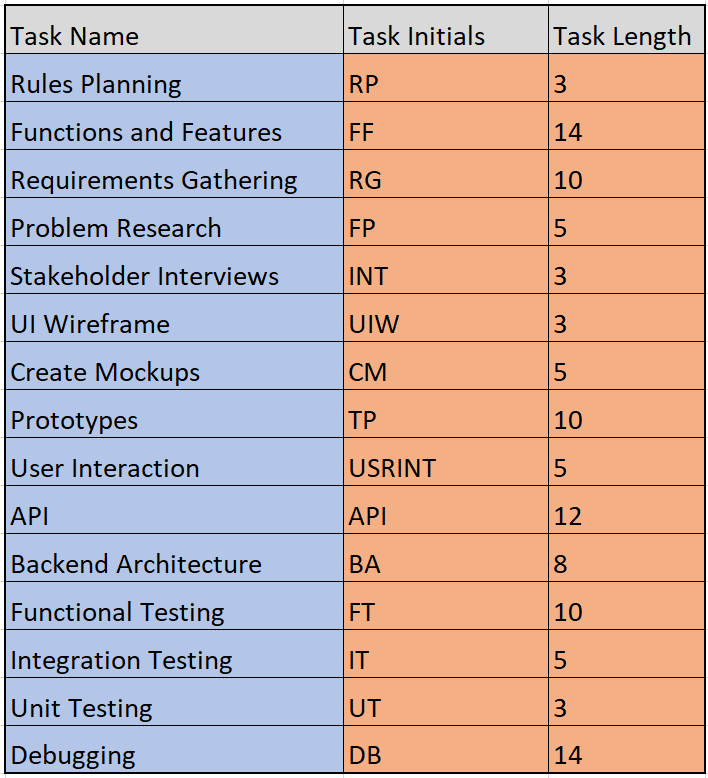
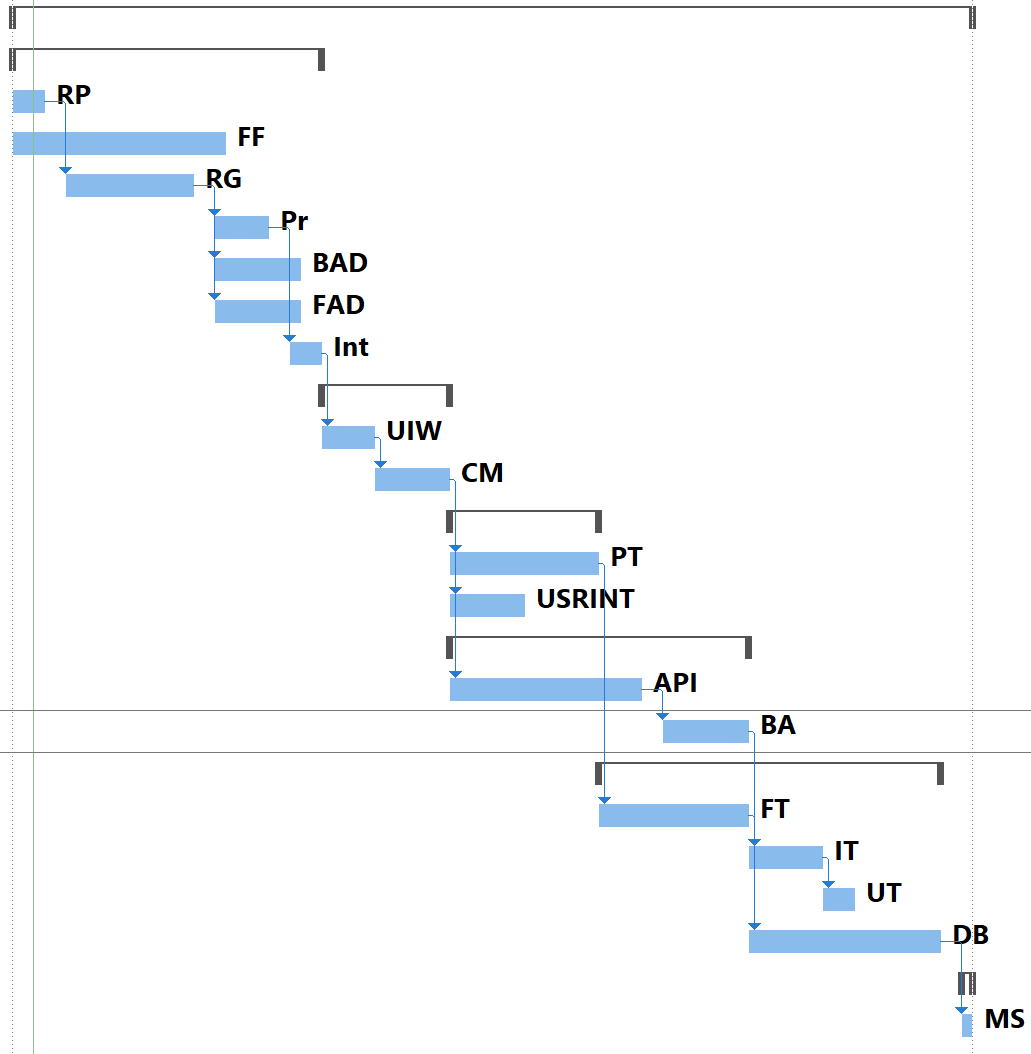
**

8.

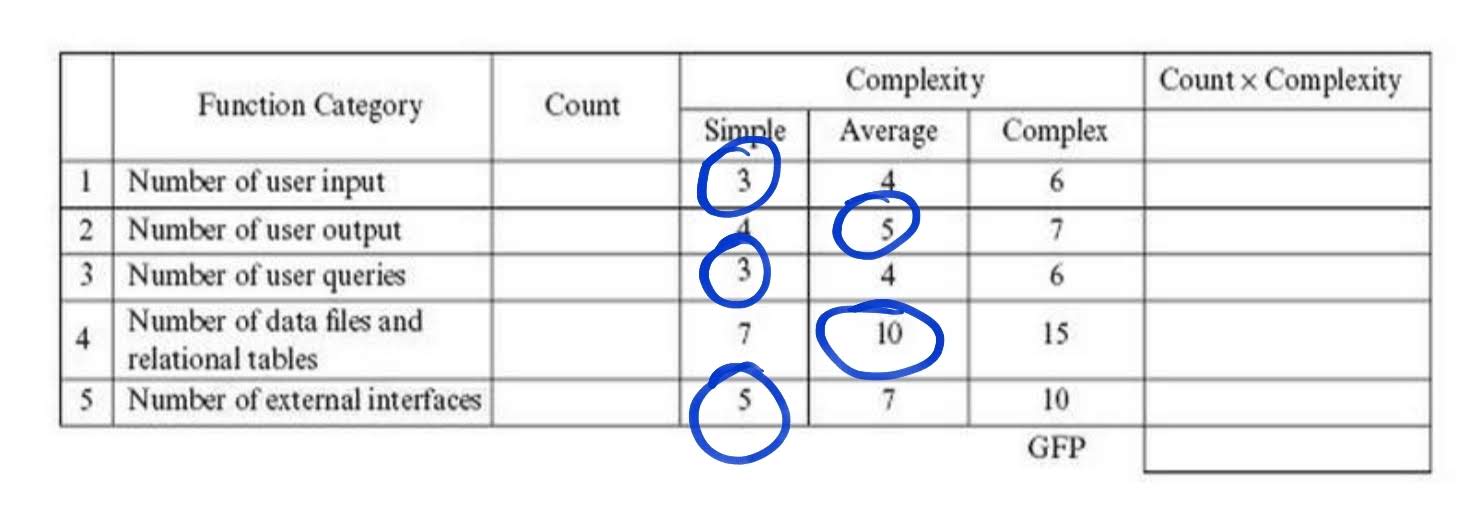


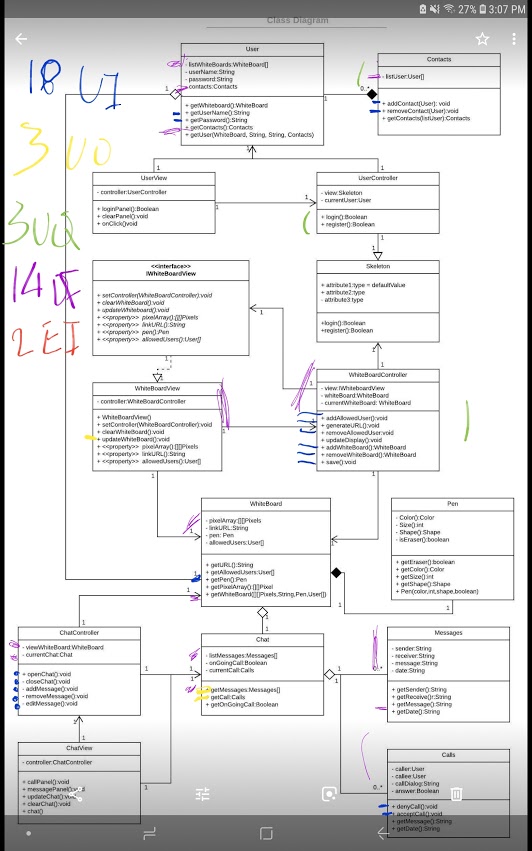
**\*\*\*END OF DELIVERABLE ONE\*\*\***

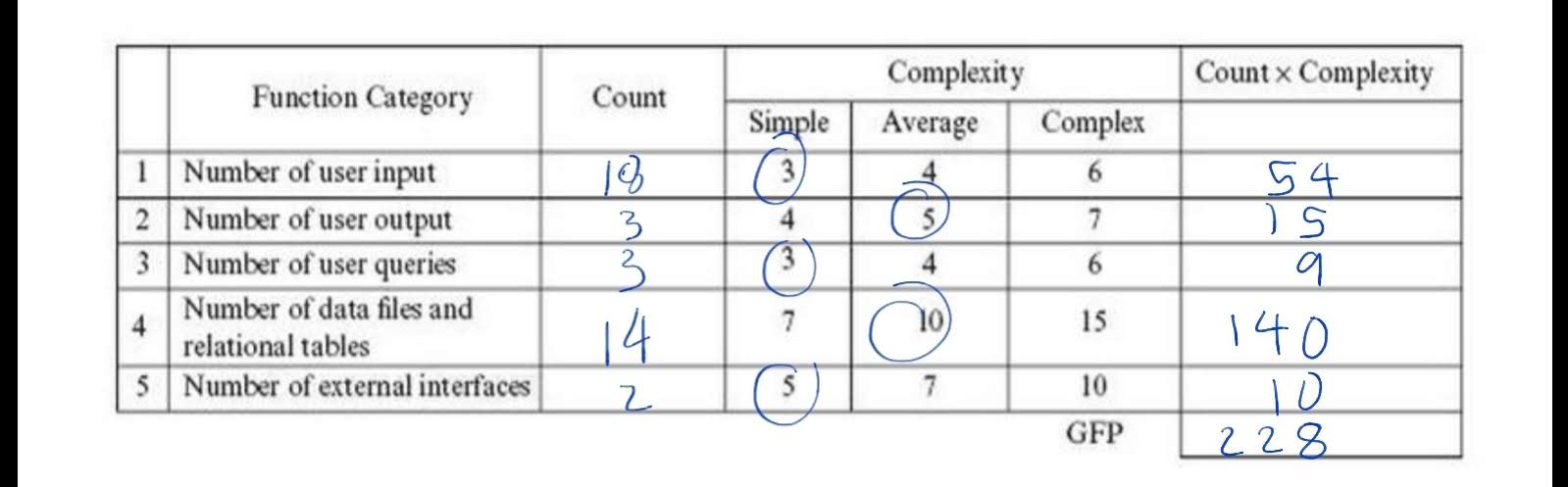
**3.1) Project Scheduling**

****

**3.2) Cost, Effort and Pricing Estimation**

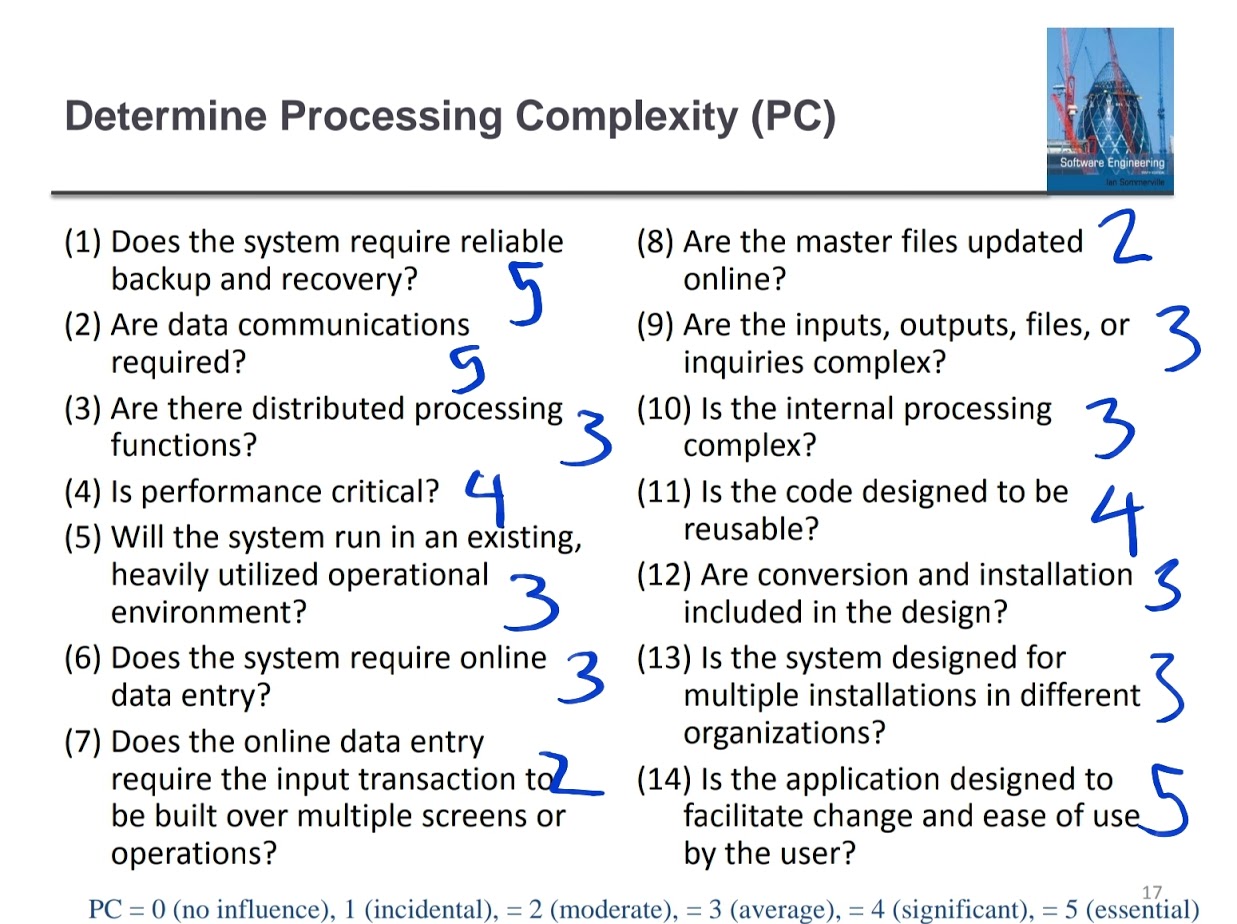




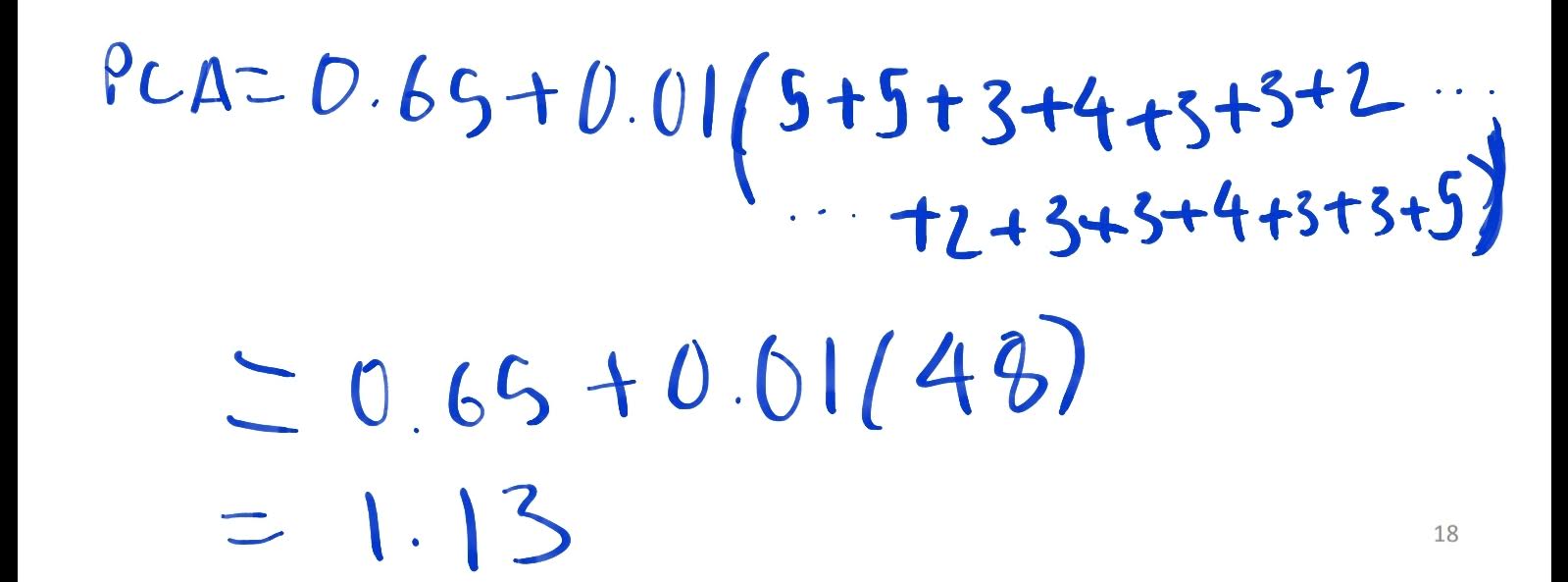
****

(Implemented Function Point)

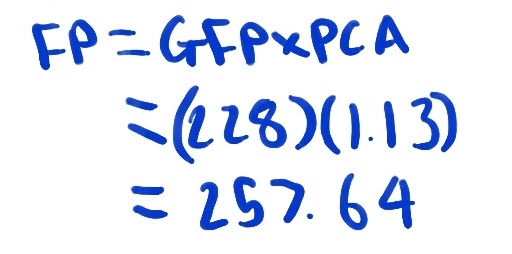
GFP = 228



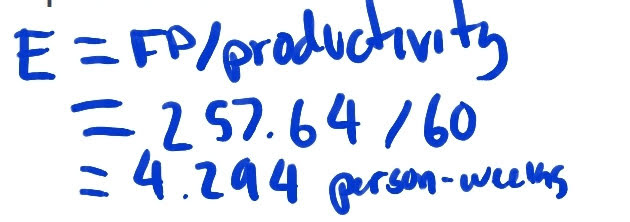
PC total = 48



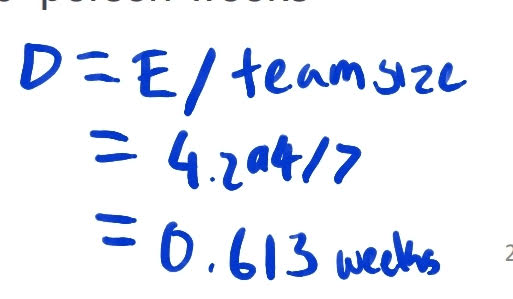
PCA = 1.13



FP =257.64

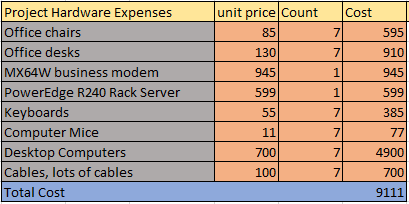


E = 4.294 person-weeks



D = 0.613 weeks

**3.3) Hardware Costs**

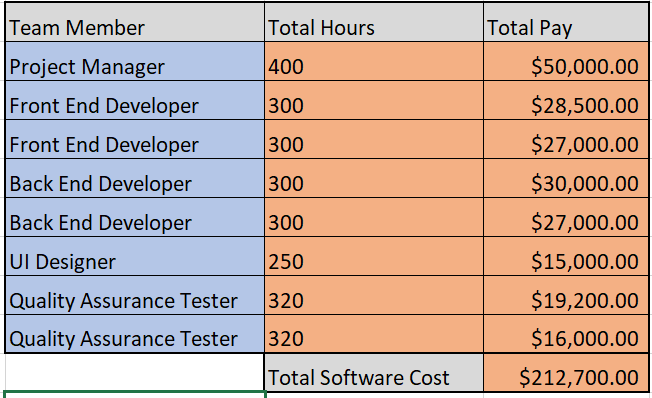


**3.4) Estimated cost of software products (such as licensed software, etc.)**

The estimated cost for backend and frontend software development for multiple platform capability is $65,610. This estimate counts the costs of software for message exchange, multimedia sharing, voice calling, contact information, notifications, registration, and settings features.

|  |  |  |
| --- | --- | --- |
| Features | Development Time (Hours) | Cost |
| Message exchange | 218 | $19,620 |
| Multimedia sharing | 40 | $3,600 |
| Voice calling | 220 | $19,800 |
| Contact Information | 72 | $6,480 |
| Notifications | 15 | $1,350 |
| Registration | 53 | $4,770 |
| Settings | 111 | $9,990 |
|  | Total Cost | $65,610 |

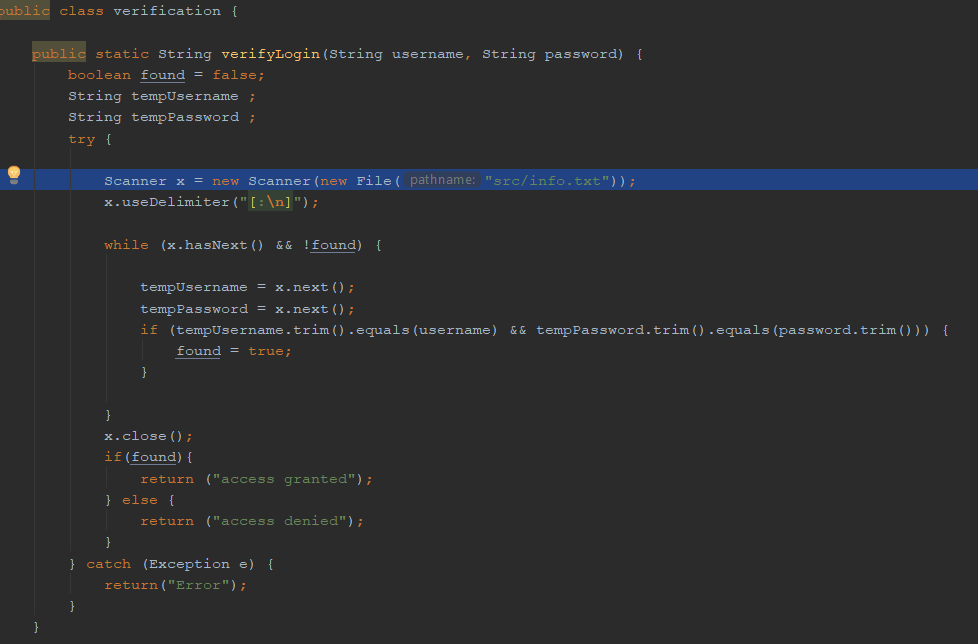
**3.5) Estimated cost of personnel (number of people to code the end product, training cost after installation)**



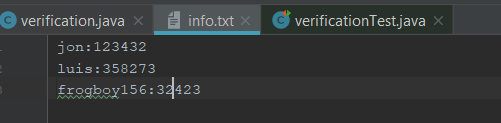
**4) Test Plan**

For our test code, we created a method that check for login verification by using the username and password and comparing it to our data that is located in our info.txt

verification.java

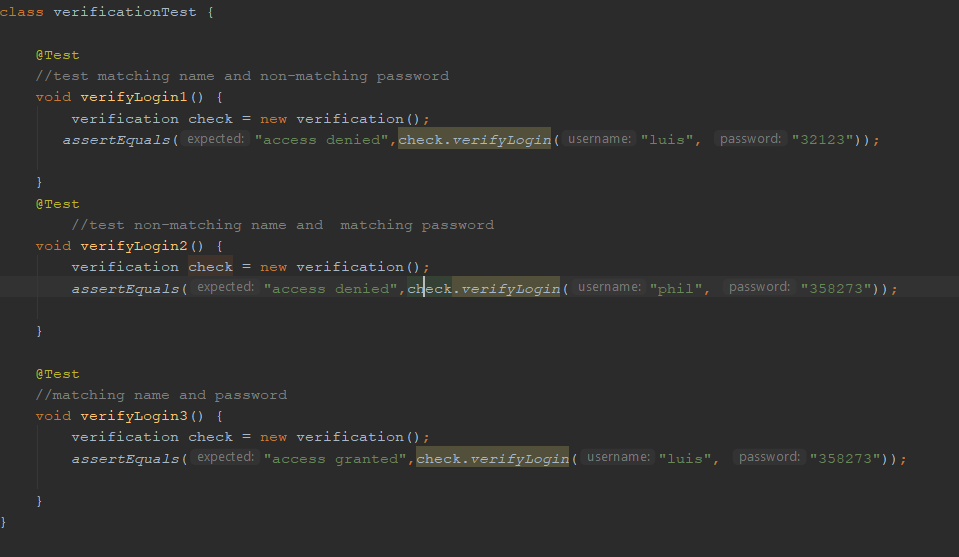


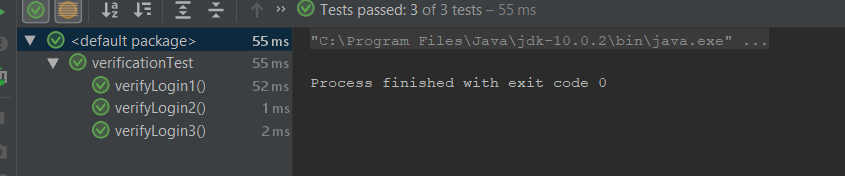
Info.txt



This is the method that compares the users input and verifies whether both username and password are present. If both cases are present our method returns the string “access granted”, otherwise it returns “access denied”.

verificationTest.java





For verfiylogin1 we test matching name and non-matching password, and use assertEquals() to check if the output equals “access denied”. Test verifylogin2 tests a non-matching name and a matching password, and checks if the output equals “access denied. Finally we test a matching username and matching password and use asserEquals() to check if the output equals “access granted”. As we see on the last image our test methods passed all our tests as we predicted.

**5) Comparison of your work with similar designs**

WhatsApp is a simple messaging app, that is similar to ours, created in 2009. It used features such as a contact list, notifications, free calls, and emoji stickers. Users were allowed to customize their wallpapers as well to make the app feel more personal. This app costed around 64k to 82k to develop. The project lasted for about 14 to 16 weeks. The development team was 5 to six people.

**6) Conclusion**

The course of this project shifted quite a bit over the course of the semester. It started out as an SMS text messenger service, but as we began making it unique different requirements were needed of us. The program shifted so that the whiteboard was the focus of the project and the messaging aspect became an extra feature rather than the main feature. To reduced costs with this change in direction we used the Model-View-Controller architecture in the program to make the program non-reliant on the device it was being run from. After that development of the project continued smoothly until completion.

**7) References**

[1] “How To Make App Like WhatsApp In 2018: Find Out Its Development Cost,” Cleveroad Inc. - Web and App development company. [Online]. Available: https://www.cleveroad.com/blog/how-much-does-it-cost-to-create-an-app-like-whatsapp. [Accessed: 19-Apr-2019].

[2] I. Sommerville, Software engineering. Harlow, United Kingdom: Pearson, 2016.

[3] Oozou, “Estimate My App,” Estimate My App. [Online]. Available: https://estimatemyapp.com/. [Accessed: 19-Apr-2019].

[4] “What would it cost to build a messaging app like Slack?,” Krit. [Online]. Available: https://builtbykrit.com/blog/how-much-would-it-cost-to-build-a-messaging-app-like-slack. [Accessed: 19-Apr-2019].

[5] “Online Home Store for Furniture, Decor, Outdoors & More,” Wayfair. [Online]. Available: http://www.wayfair.com/. [Accessed: 20-Apr-2019].

[6] Meraki, Inc, “Manage your entire networkfrom a centralized dashboard.,” Meraki. [Online]. Available: https://meraki.cisco.com/. [Accessed: 20-Apr-2019].

[7] “Dell United States Official Site | Dell United States,” Dell United States Official Site | Dell United States. [Online]. Available: http://www.dell.com/. [Accessed: 20-Apr-2019].

[8] “Computers and Electronics,” Micro Center. [Online]. Available: http://www.microcenter.com/. [Accessed: 20-Apr-2019].

**8) Presentation Slides**

