**CS673 Software Engineering** 

**Team 1 - Project Name**

**Project Proposal and Planning**

|  |  |  |  |
| --- | --- | --- | --- |
| Team Member | Role(s) | Signature | Date |
| Wasupol Tungsakultong | Team Leader | *WT.* | 09/10/2020 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Change** |
| **1.0** | Team 1 | **09/10/2020** | **09/22/2020** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[Overview](#_87t9hln2vjz0)

[Related Work](#_mps353x5ezyl)

[Detailed Description](#_fg3z0hpd4q9v)

[Management](#_ds8oyr75pnh1)

[Plan](#_ds8oyr75pnh1)

[Process Model](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.27177f40uci)

[Risk Management](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.a4oqwntk3mw)

[Monitoring and Controlling Mechanism](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.ywdoc2clc9yt)

[Schedule and deadline](#_tadq5mb0pici)

[Quality Assurance Plan](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.72e1f4uawy2r)

[Metrics](#_b2haznn3yyz2)

[Standard](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.vc72k6dweldv)

[Inspection/Review Process](#_f1c69ifi68h7)

[Testing](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.r5d5mhtlf0kq)

[Defect Management](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.54a4wuncjg1c)

[Process improvement process](#_jhct37ebxxpn)

[Configuration Management Plan](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.hw41vg4ykxen)

[Configuration items and tools](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.bwlb4d4vdox2)

[code commit guidelines](#_yyauft6zr9hw)

[References](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.8mva2050iy7t)

[Glossary](#_ty3i2nqffhtc)

# Overview

(Please give an overview of your project. It should include the motivation, the purpose and the potential users of the proposed software system. )

Social and relationship have never been overemphasized. The Internet is supposed to increase social interaction and strengthen social connections. Unexpectedly, it greatly weakens the ability of interpersonal communication in real life. We have lost the confidence to build a loving relationship. Although the existing social media is designed to help individuals build open and diverse relationships, we still feel lonely, and we still long for intimacy, privacy, and communication in an inner circle. Who didn’t refuse the Facebook invitation from parents? Who doesn't have some ideas which they only want to share with their close friends? This app is intended to help improve the bridge between The intended audience for this system is catered to high school students, university students, and young professionals.

# Related Work

(Please describe any similar software systems that you have found through the online research, and the differences between your software and those software systems.)

[Join Me (Video sharing screen on mobile application)](https://www.join.me/resources/user-tips/how-to-share-screen-with-android)

[Slack](http://slack.com/)

[Whatsapp](https://www.whatsapp.com/)

[Facebook Messager](https://www.messenger.com)

# Proposed High level Requirements

* 1. Functional Requirements  
     (For each functional requirement, please give a feature title and a brief description using the following format: As (a role), I want to (action), so that (value).)
     1. Essential Features (the core features that you definitely need to finish):

(For each essential features, please give a rough estimation in terms of person hours or an range of person hours)

1. As a user, I want to send a message to others, so that I can have a private chat with my friend.
2. As a user, I want to send pictures to others, so that I can share my beautiful pictures to my friend.
3. As a user, I want to login to the application with my social media, so that this will make me more comfortable using this application.
4. As a user, I want to create a chat group of my close friends, so that I can have my close friends discussion.
5. As a user, I want to send a file or location to my friend, so that I can share important files or location for meeting up.
6. As a user, I want to invite other friends to my current chatroom, so that I can carry on my topics when my friends.
7. As a user, I want to receive a notification when my friends message me, so that I can reply to them instantly.
   * 1. Desirable Features (the nice features that you really want to have too):
        1. As a user, I want to have a SMS Authentication, so that I can ensure more security.
        2. As a user, I want a Google Translate in the chat, so that I can understand my friends who are foreigners.
        3. As a user, I want to see a feed or story of my friends that they are sharing, so that I can know what they are interested in.
        4. As a user, I want to follow my friends, so that they can follow me back and see my feed.
        5. As a user, I want to have status, so that my friends will know that I am available or away. (AIM style away message)
     2. Optional Features (additional cool features that you want to have if there is time):
        1. As a user, I want to send stickers to the chat.
        2. As a user, I want to have video screen sharing on mobile.
        3. As a user, I want to have my own profile.
        4. As a user, I want to backup my history chat
   1. Nonfunctional Requirements
      * 1. Google Analytics (Track which features users use the most.)
        2. Set up automation build deploy
        3. Performance monitoring
        4. Encryption and security (both android, backend)

# Management Plan

## Process Model

(Please describe your software process model, e.g. ?)

The project uses Agile methodology with a partially scrum framework. The team leader will be the role of scrum master and task planner. One the first day of the week, It will be the planning meeting with scrum poker will all the members. Every member of the team will give a score of how hard of the task, priority and estimation.

There will be a scrum meeting in the evening to update the current status of the task and what will do for the next day, but it is not a standup meeting like usual. There will be a scrum retrospective at the end of the week to provide feedback, problems and improvement for each iteration.

## Objectives and Priorities

(Please describe your project objectives with highest priority first. Project Goals can include but not limited to complete all proposed (essential) features, deploy the software successfully, the software has no known bugs, maintain high quality, etc )

We would like to ensure that the chat application maintains high quality throughout its usage. From a user perspective, a measure that can be used to assess the quality is the amount of failed chat applications per a reference chat number. A fail can be defined as a crashed chat, incorrect message delivery to a recipient, and having a message send/receive time within an acceptable time window. We would ideally like to have this metric to be greater than 80%. From a development perspective, we would like to maintain the amount of defects around industry standard. Therefore, the project is to maintain approximately 1 - 10 defects/1000 lines of code (<http://web.mit.edu/6.031/www/sp19/classes/03-testing/>). All essential features identified in the requirements should be implemented. These are listed in section 3.a.i.

## Risk Management (need to be updated constantly)

(Please write a summary paragraph about the main risks your group identified and how you plan to manage these risks. Then use the separate google sheet for detailed risk management. The template is provided in the same folder with this file. Please provide the link to the sheet.)

Risk Management Sheet Link: [Risk Management of Team 1](https://drive.google.com/file/d/1JP1qozY6WrRQtbDxUnDIyEhB3upHGHQC/view?usp=sharing)

## Monitoring and Controlling Tools and Mechanisms

We will use the following tools to facilitate group communication and monitor the project progress.

* + 1. Pivotaltracker Link:
       1. [Pivotal Tracker Team 1](https://www.pivotaltracker.com/n/projects/2465407)
    2. Slack Link:
       1. [Slack chat](http://bumetcs673f20.slack.com)
    3. Github Link:
       1. [Github Team 1](https://github.com/BUMETCS673/CS673F20T1)
    4. Zoom meeting Link:
       1. [Team leader's Zoom link](https://bostonu.zoom.us/j/95008191650?pwd=RUtSS2tTbnFGeUpya0VTWXlXZXBIZz09)
    5. Weekly meeting time:
       1. Sunday morning @ 11 - 12:00 pm EST.

## Timeline (need to be updated at the end of each iteration)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration | Functional Requirements(E/D/O) | Tasks | Estimated/real person hours | Presentation Recording Link (5-10 minutes) |
| 1 | E | 6 | 24/- |  |
| 2 | E+D | 7 | 24/- |  |
| 3 | D+O | 5 | 24/- |  |

# 

# Quality Assurance Plan

## Metrics

* + 1. Definition (e.g. define what metrics will be used, , how to keep track of metrics, and how to analyze the metrics for process improvement. Two types of metrics should be included: product metrics and process metrics. Particularly include product complexity (LOC, # of files, # of classes, # of methods etc.) cost (in terms of man hours), defect and defect fix rate etc.user story points,
    2. Results (to be completed at the end of each iteration),
    3. Product metrics:
       1. Defects per lines of code; choice for complexity measurements will be revisited
       2. Amount of failed chat applications per a given reference number of chat applications (chose for reference number to be determined)
    4. Process metrics:
       1. Defects resolved per an iteration/per man hours
  1. Standard  
     (e.g. documentation standard, coding standards etc. )
     1. Implementation Kotiln base on standards by JetBrain [Kotlin Code Standard](https://kotlinlang.org/docs/reference/coding-conventions.html)
     2. Implementation Java base on standards by [Oracle](https://www.oracle.com/java/technologies/javase/codeconventions-contents.html)

## Inspection/Review Process

* + 1. Gitflow workflow: At least one person will review the develop branch code prior to release and the master branch. Introduction to Git Flow [here](https://www.git-tower.com/learn/git/ebook/en/command-line/advanced-topics/git-flow) [here](https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow).
    2. Timing: Feature development will be planned for the first two weeks of the iteration. The last week of the iteration will focus on testing, maintaining and defect management.

## Testing:

Testing Results Doc Link: [Testing Results Team 1](https://docs.google.com/document/d/1M-B-zFN0D2fA7_4cteE6AjrKUcHEWMnUHHLN6JvvMbw/edit)

* + 1. Unit testing: Every developer will write unit testing code.
    2. UI testing: Done by QA Leader at the end of each iteration?
    3. Other types of testing (integration, functional, non-functional) will be performed at the end of each iteration.

## Defect Management

* + 1. Management tool: track via PivotalTracker (story type = bug)
    2. Severity
       1. Critical: Issue causes immediate crashes and prevents app from running. Fixing this type of defect will be given the highest priority.
       2. High: Intermittent crashes and/or major issues with the usability of the main features of the app. This level of severity will need to be addressed prior to the completion of the iteration.
       3. Medium: App continues to run but feature testing yields unexpected or incorrect results. This level of defect will not prevent the completion of the iteration but will instead be moved to the backlog for the next iteration (if it is not fixed in the current iteration).
       4. Low: Minor impact to usability, core app features unaffected. Users can work around the issue. Like medium severity defects, these defects will not prevent the completion of the iteration.
    3. Priority
       1. High: Critical defects. Need to be fixed as soon as possible, ideally within a day and before the end of the iteration.
       2. Medium: High/Medium severity. Depending on severity, this may need to be addressed before the end of the iteration.
       3. Low: Low severity. Can be added to the backlog.

# Configuration Management Plan

(For more details, please refer to SCMP document for encounter example)

## Configuration items and tools

* + 1. [Fastlane](https://fastlane.tools/) is for setting Android application environments.

## Change management and branch management

* + 1. Using Git Flow by [Vincent Driessen at nvie](http://nvie.com/posts/a-successful-git-branching-model/) for git branching strategy and management

## Code commit guidelines

* + 1. Code commit base on Gitflow and Git message commit will follow standard [Git commit best practices](https://gist.github.com/turbo/efb8d57c145e00dc38907f9526b60f17)
  1. Integration and deployment plan
     1. All working code will work on `Feature`, `Support` and, `Hotfix` branches before merging to the develop branch which has to pass all essential unit and ui tests. By merging to release branch, The application will be built, compiled and archived by [CircleCI](https://circleci.com/) using [Fastlane](https://fastlane.tools/) command, then It will be released on [AppCenter](https://appcenter.ms/).

# References

(For more details, please refer to the encounter example in the book or the software version of the documents posted on blackboard. )

1. Vincent Driessen. 2010. A successful Git branching model: [https://nvie.com/posts/a-successful-git-branching-model](https://nvie.com/posts/a-successful-git-branching-model/)

# Glossary