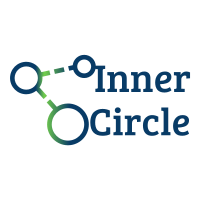
**CS673 Software Engineering** 

**Team 1 - Inner Circle**

**Project Proposal and Planning**

|  |  |  |  |
| --- | --- | --- | --- |
| Team Member | Role(s) | Signature | Date |
| Wasupol Tungsakultong | Team Leader | *WT.* | 09/10/2020 |
| Jian Ma | Backup Leader | *JM* | 09/10/2020 |
| Peter Lake | QA Leader | *PL* | 09/24/2020 |
| Anaya Sharma | Requirement Leader | *AS* | 09/24/2020 |
| Michael Ranasignhe | Design and Implementation Leader | *MR* | 09/24/2020 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Change** |
| **1.0** | Team 1 | **09/10/2020** | **09/22/2020** |
| **1.2** | Team 1 | **10/01/2020** | **12/10/2020** |
|  |  |  |  |
|  |  |  |  |

Content

[Overview](#_heading=h.2xcytpi)

[Related Work](#_heading=h.1ci93xb)

[Detailed Description](#_heading=h.3whwml4)

[Management](#_heading=h.2bn6wsx)

[Plan](#_heading=h.2bn6wsx)

[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](#_heading=h.qsh70q)

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

[Metrics](#_heading=h.3as4poj)

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

[Inspection/Review Process](#_heading=h.1pxezwc)

[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](#_heading=h.49x2ik5)

[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](#_heading=h.3j2qqm3)

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

[Glossary](#_heading=h.4i7ojhp)

# Overview

Social and relationship have never been overemphasized. The Internet is supposed to increase social interaction and strengthen connections. Unexpectedly, it greatly weakens the ability of our interpersonal communication in real life. We have lost the confidence to build a loving relationship and lost the incentive to connect with individuals and nature. Although the existing social media is designed to help individuals building open and diverse relationships, we still feel lonely, and we still long for intimacy, privacy, and communication with our inner circles such as our close friends and loved ones. 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? Who didn’t have friends who set their Instagram account as private?

It is time to change it! We're not living in ways that meet our deepest needs. We want to share our daily life stories, we want to feel the same feel of our close ones, we want to be connected. Yes, connection!

This app is intended to build a community with only our inner circles. We share our readings, music, opinions, and daily stories freely and no need to worry about the audience. Here, we are a concrete person instead of a mixture of tags, titles, or fancy pictures. Let's start to create your own warm, cozy, trusted community!

Based on the tone of this product, the intended audience is catered to high school and college students, urban youth, and young professionals.

# Related Work

* 1. [Facebook](https://play.google.com/store/apps/details?id=com.facebook.katana&hl=en_US)
     1. Similarities
        1. Users build connections most from their real life.
        2. Users can post updates with texts, links, pictures, etc.
        3. Users can comment under a post.
        4. Users can have real time chat with their friends.
     2. Differences
        1. The tone of Inner circle is younger and chicer than Facebook

According to data from Ampere Analysis, approximately 44% of Snapchat users are between 18 and 24 years old, while only 20% of Facebook users are in this critical age group. As more and more older users join, the mass departure of young people has been balanced. At this time, Facebook's first users have reached 30-40 years old.

* + - 1. Inner circle is always online.

Instead of showing online or offline status, and messages being read or not, Inner circle does not recognize the status of users and messages. Users send messages at any time they want, same as the responders.

* 1. [Facebook Messenger](https://www.messenger.com) & [Whatsapp](https://www.whatsapp.com/)
     1. Similarities
        1. Users build connections most from their real life.
        2. Users can have real time chat with their friends.
        3. Users can have video calls and audio calls.
        4. Users can build a group chat with their friends.
     2. Differences
        1. Users can share their feelings, opinions, daily stories, videos, etc. in the Broadcast block.

Inner circle is not a pure chat application. It is a place for people to communicate their thoughts and feelings with their friends, it is a community for people to care and love each other.

* + - 1. Inner circle is always online.

Instead of showing online or offline status, and messages being read or not, Inner circle does not recognize the status of users and messages. Users send messages at any time they want, same as the responders.

3. Video screen sharing

Users can share their own screen with the cursor during video calls. A productive feature for users to discuss with a specific document anywhere anytime.

* 1. [Instagram](https://apps.apple.com/app/instagram/id389801252?vt=lo)
     1. Similarities
        1. Users can chat with their friends on this platform.
        2. Users can post updates with pictures or videos.
        3. Users can comment under a post.
     2. Differences
        1. Instagram users are most weak connected, while Inner Circle users are strong connected.

Instagram is more about sharing to the general public, and it provides only chat rooms for users. The connection between users is weak. Inner Circle focuses more on communication between users, providing video calls, file transmissions, etc. The main purpose is to facilitate connections between each other. It is more like a lovely neighborhood.

* + - 1. Instagram focuses on socializing with the public, while the Inner Circle focuses on social interaction with close ones.

Instagram is facing public. Users are not only care of sharing, also care of traffic, and influence. While Inner Circle is facing friends in reality, users care more about conversation and communication by sharing their feelings and real life moments, rather than something like is this sharing fancy, chic, and beautiful or not.

# Proposed High level Requirements

* 1. Functional Requirements
     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 account, 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.
8. As a user, I want to add my friends with QR code, so that I can have my friends in contact list and build closer relationships within the application.
   * 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. As a client, I want to track which features users use the most by adding Analytics.
        2. As developers, I want to set up automation building and deployment, so that program will reduce failures and bugs.
        3. As testers, I want to add performance monitoring in the application, so that I will know and predict the application will be a good experience for users.
        4. As a security leader, I want to add encryption to both android and backend.
        5. As a security leader, I want to have end to end encryption for my correspondence, so that I can ensure that users’ messages are personal and private to them.

# Management Plan

## Process Model

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

The project uses Agile methodology with a partial 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 and 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 daily standup meeting in the evening to update the current status of the task and what will be done 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 chats 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)

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. Monday evening @ 10 - 12:00 am 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 | 24/171 | [Presentation](https://bu-metcollege.zoom.us/rec/play/SR1TUI8cf6pILEPuys0i2hBGTWZCfRLMBhwUoAZdVWWHpnFw5_oEqQ88-PP6xtr7a3vthDQMd09jZP63.ysABG9lBvP2C6y6P) |
| 2 | E+D | 7 | 120/154 | [Presentation](https://drive.google.com/file/d/1X6brBSUoJPZGSv4E1lZiT7BCxSGinMS3/view?usp=sharing) |
| 3 | D+O | 5 | 120/202.5 | [Presentation Demo](https://drive.google.com/file/d/1X6brBSUoJPZGSv4E1lZiT7BCxSGinMS3/view?usp=sharing) |

# 

# Quality Assurance Plan

## Metrics

* + 1. Product metrics:
       1. Defects per lines of code; choice for complexity measurements will be revisited
       2. Amount of failed chat messages per a given reference number of chat messages (choice for reference number to be determined)
    2. Process metrics:
       1. Defects resolved per an iteration
       2. Man hours per a defect resolution
  1. Standard
     1. Documentation:
        1. SPPP: To continually refine the scope of the project
        2. Progress Reports: Tracks individual contributions to the project and informs management plan
        3. Risk Management: Continually tracks and addresses risks
        4. Meeting Minutes: Reference for team members responsibilities and focusses in a given week
        5. SDD: Software Design Document, focus on technical scope of the project.
        6. STD: Software Test Description, provided by QA leader, tests’ scope that cover main functionality of the project.
     2. Coding Standards:
        1. Implementation Kotiln based on standards by JetBrain [Kotlin Code Standard](https://kotlinlang.org/docs/reference/coding-conventions.html)
        2. Implementation Java based 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) [h~~ere~~](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 an 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 [Github Actions](https://github.com/features/actions) using [Fastlane](https://fastlane.tools/) command, then It will be released on [AppCenter](https://appcenter.ms/), while Backend code will be deployed on [AWS](https://aws.amazon.com/).

# 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/)
2. MIT Software Construction. Lecture 3

<http://web.mit.edu/6.031/www/sp19/classes/03-testing/>

# Glossary