**Zero Feature Release**

**Human ICT Software Engineering \_ Team CCC**

**Hong Jin, Park Juhyeon, Kim Kyoungchan**

**Choi Jonghyun, Rajat, Antonie**

**Contents**

1. **Documentation for Users**
   1. A description of your project p.3
   2. How a user obtains and install the software (including prerequisites) p.3
   3. How to run the software p.4
   4. A high-level description of the user view of your software p.4

1.5 How to user the software p.5

1.6 How to report a bug p.6

1. **Documentation for developers**
   1. How to obtain the source code p.8
   2. The layout of your directory structure p.8
   3. How to build the software p.9
   4. How to test the software p.9
   5. How to set up an automated daily (or more frequent) build and test p.10
   6. How to release a new version of your software p.10
   7. How to access the list of outstanding bugs and the list of resolved bugs, and how to resolve a bug p.11
   8. The source code should contain comments p.12

**Zero Feature Release**

Team CCC

Report Version 3.0

**<Documentation for Users>**

1. **A description of project**

* CCC is a messenger app like any other messenger app especially developed for University Student’s but it has some added functionalities which can be very useful for managing day to day activities.
* It comes with a scheduler which can access your university time table and show you when you have classes or not and lets you to even manage your individual schedule, added to this you can also use schedule sharing function which lets you to share your schedule over the messenger so that you can arrange your meeting with your friends without going over calendar again and again.
* So basically, with this app you can talk with your with friend whenever and wherever you want, manage your personal schedule, and share your schedule with your friends.

1. **How a user obtains and install the software(including prerequisites)**
2. Requirement

* Smartphone with API Level >= 22 (LOILIPOP)

Because our project based on it.

* Gapps (Google Application suit)

1. How to install

* It will be a very simple process.
* You are done searching the Google Play Store for “CCC” and pressing the Install button. That`s All

1. **How to Run the software**
2. Running the software is a simple task.
3. Simply click on our installed application and run it!
4. We will require a number of privileges to ensure smooth execution.

* Authority for photo album
* Authority for photo album
* Authority for address
* Authority for Internet access

1. **A high-level description of the user view of your software**
2. Useful UI

* UI is simple, and it is easy to search what the user wants.

1. Main page

* The user signs up first and then logs in with his / her ID and password.
* The main page provides various functions such as timetable, scheduler, and chatting buttons for each room. It also allows users to view the weekly scheduler, which is a collection of users' schedules and timetable.

1. Chatting

* The user opens a chat room to use chat. This will create a unique code, and if other users enter this unique code, they can share the chat room.

1. Timetable

* The user finds the subject corresponding to his / her course among all the courses opened at the CAU and adds it to his/her timetable.
* User can add subjects by pressing the + button on the timetable screen.

1. Scheduler

* The user enters his schedule by clicking the + button on the desired date.
* The user can see over several months' schedules.

1. **How to use the software**
2. Sign Up

* You must sign up first to use our application.
* You can enter your ID, Password and additional information.

1. Log In

* Sign in to CCC by entering the ID created by Sign Up.
* The first time you sign in to a device, you’ll stay signed in unless you sign out.

1. Register Chat Room

* In order to use messenger, it is necessary to register chat room.

1. Make Chat Room

* Pressing the Make Chat Room button will create a unique number for that chat room, which will allow you to share that chat room with others.
* So if you have a unique number in your chat room, you can use two or more people.

1. Send Message

* After entering the desired message, enter the Send button.
* You can send messages with enough content regardless of the number and length.

1. Send Photo

* Select the desired photo in the photo tray and press the Send button to send the picture.

1. Register Time table

* Find the course you want by entering the department, subject classification, subject name, and so on.
* If you select a subject through the above process and add it, it will be automatically registered in the timeline.

1. Delete Time table

* When you select a subject in the timetable and click the Delete button, the subject is deleted.

1. Import Time table

* Get timetable backed up via file input.

1. Export Time table

* Export the time table to a file for back up.

1. Register Schedule

* Click the appropriate date in the calendar like scheduler.
* Then enter the appropriate schedule at that

1. Delete Schedule

* Select the date and time and press the Delete button to delete it.

1. Import Schedule

* Get timetable backed up via file input.

1. Export Schedule

* Export the time table to a file for back up.

1. Share Weekly Schedule

* When you press Sharing Weekly button on the Messenger Screen, Weekly Scheduler on the main screen is automatically sent to the other party.

1. **How to report a bug**
2. Data-driven Processing

* If an error occurs at each location, check first for the cause of the error.
* Automatically tells the developer where the error occurred, but if there are any additional things you want to add, it will be a good reference.

1. Periodic Feedback

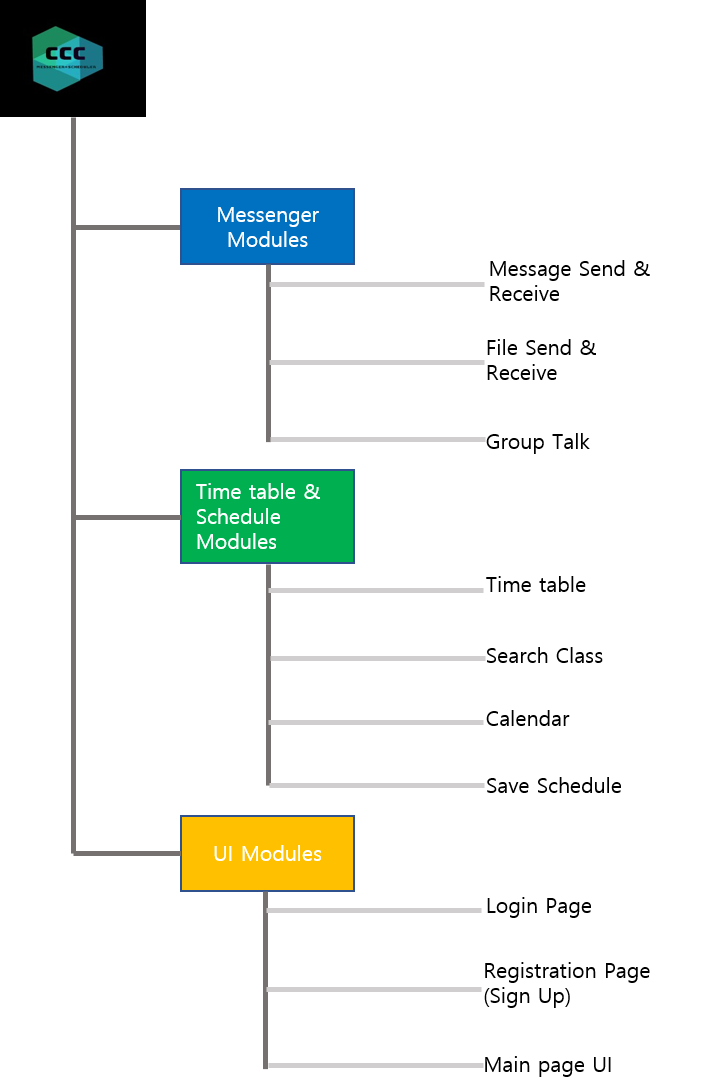
* We are going to check the frequency of the bug based on the data weekly or monthly graph.
* We will also monitor to see if the bug is again bugged after the bug has been debugged.

**<Documentation for Developers>**

1. **How to obtain the source code**

* If you want the project source code, send the mail to anyone on the ccc team and provide the source code.
* Using Git, each developer saves their code, integrates them, completes one program, and stores it in the git hub. Those who need the program source code can go to the git hub and download the source.

1. **The layout of your directory structure**



1. **How to build the software**

* We are developing software using the Android studio.
* We divided software into three parts: messenger, scheduler, UI and then carried out each part by two people.
  + First, the contents of the project were defined, and the overall plan, personal plan were determined.
  + Second, Design and detail were added according to the overall plan, and the layout of each part was done according to personal plan.
  + Third, we are currently working on firebase integration and customer requirements, and we will make test cases and test it.

1. **How to test the software**

* We will test with our android smart phones and android virtual machine which provided in android studio.
* With these devices, we can test lots of things like:
  + Detect a bug in running application.
  + Check a design is correct we thought as.
  + Demonstrate it meet the requirements.
* First, we can detect a bug in running application using android devices. When we click a button, type a text, touch somethings in application, we can check a reaction of our behavior is correctly functioned. Otherwise, if something incorrectly reactioned, we think this function has a bug and we should fix this situation.
* Second, we can check a design is correct we though as. In test situation we can watch design of application and check things like buttons, textboxes are in correct position. We also can watch design of activity in android studio when we implement xml files.
* Third, we can demonstrate our application meet the requirements. When we test application, we can test whole things of applications. Because android studio will make an application which we implemented.

1. **How to set up an automated daily (or more frequent) build and test**

* We use android studio which have automated build tool.
* This is gradle file that has information about the way to build.
* So we don’t have to set up makefile and we just set up the gradle file.
* We will set gradle file with api level 22: ‘lolipop’ for many people can use our application.
* We will upload whole things of our application in github and build gradle file also can be seen in our github.

1. **How to release a new version of your software**

* We use git to manage our project processes.
* So we will use git to release the software and it will help us to manage version of released software.
* If we use git to release our software, we can gain things of benefits.

1. We can easily rollback process if we got serious problems. One of main function of git is that we can manage version of software. We can see all things of our released version of software and we can rollback to any version of software.
2. We can do recover process with any version of software. If we got serious problems and it can be recover at order version of software, we can recover process using order version of software.

* This is why we use git to release a software.

1. **To Find the Bugs and to resolve them**

* We can use Android built in Debugger to debug our app.
* We can write System Logs while debugging our app. Log messages help us understand the execution flow by collecting the system debug output while we interact with your app. Log messages can tell us what part of your application failed.
* We can use Log information to identify the bugs or points where our program failed and then we can correct our code where it fails.
  + We Can also use Breakpoints
* Android Studio supports several types of breakpoints one of them is a line breakpoint that pauses the execution of app at a specified line of code. While paused, we examine variables, evaluate expressions, then continue execution line by line to determine and identify the causes of runtime errors
  + We run different Test Cases on our app to identify various bugs. For this we prepare good test cases for our environment and we stressed on functional test cases which include major risk of the application.
  + If our apps fail at any test then we try to identify and resolve the problem which caused the test to fail.

1. **The source code should contain comments**



* As you can see in the picture above, each code is annotated to make it easier to develop by showing what the code means in an environment developed by various developers.