



2016

MOHAMED265

1



**Cairo University**

**Faculty of Computer and Information**

**Computer Science Department**

# Find My Device

**Supervisor**

**Ahmed Shawky Mousa**

**Team**

Mohamed Fouad	01007523776	20120334
Ibrahim Ali	01227462847	20120004
Amany Mohamed	01148842970	20120105
Ahmed Yehia	01110949585	20120071
Neama Fouad	01115665358	20120426



# Table of contents

---

1. Introduction	3
1.1 Introduction.....	4
1.2 Motivation.....	4
1.3 Problem definition .....	4
1.4 Project objective.....	4
1.5 Gantt chart of project.....	5
1.6 Project development methodology.....	11
1.7 Report organization.....	18
2. Related work	19
3. System analysis and design	22
3.1 Project specification	
3.1.1 Functional requirement	
3.1.2 Non-functional requirements	
3.2 Use case diagrams	
3.3 Sequence diagrams	
3.4 System test phase.....	51
3.5 Suggested solution.....	56
3.6 Primarily design and architecture.....	57
4. Screenshots	69
5. Summary	87
References	89
Appendix Code Links	90

## List of figures

---

Figure 1 .....	6
Figure 2 .....	7
Figure 3 .....	8
Figure 4 .....	9
Figure 5 .....	10
Figure 6 .....	57
Figure 7 .....	58
Figures 8 class diagram .....	59
Figures 9 .....	70

## List of tables

---

Table 1 .....	11
Table 2 .....	12
Table 3 .....	12
Table 4 .....	13
Table 5 .....	13
Table 6 .....	14
Table 7 .....	14
Table 8 .....	15
Table 9 .....	15
Table 10 .....	16
Table 11 .....	16
Table 12 .....	21

# Chapter 1

## Introduction

## **Introduction:**

---

We help people to access their lost or stolen electronic devices like tablet, mobile and laptops, via web app and pre-installed app on these devices.

Even if this device wasn't lost, its owner can access it remotely via our web application.

This access includes operations like (access file system storage, get file from the device, delete file, know the location of device, record voice, record video).

## **Motivation:**

---

As we live in modern era of speed and technology, everyone seeks to have a quick access to information while they possess multiple devices containing different data. The user now become in a need of a new way to have all this information instantly wherever it is in which device it is. By that, his life routine will be much simpler. Furthermore availability of the Internet everywhere and at high speed and low cost will change the shape of life, so we think that building our Graduation Project will facilitate the communication with your devices even you are away.

## **Problem definition:**

---

We help people to access their lost or stolen electronic devices like tablet, mobile and laptops, via web app and pre-installed app on these devices..

## **Project Objective:**

---

Developing system which mange you to find your devices, retrieve files from uncontrollable devices, determined location, record sound and video via web browser and pre-installed client on device.

## Gantt chart of project

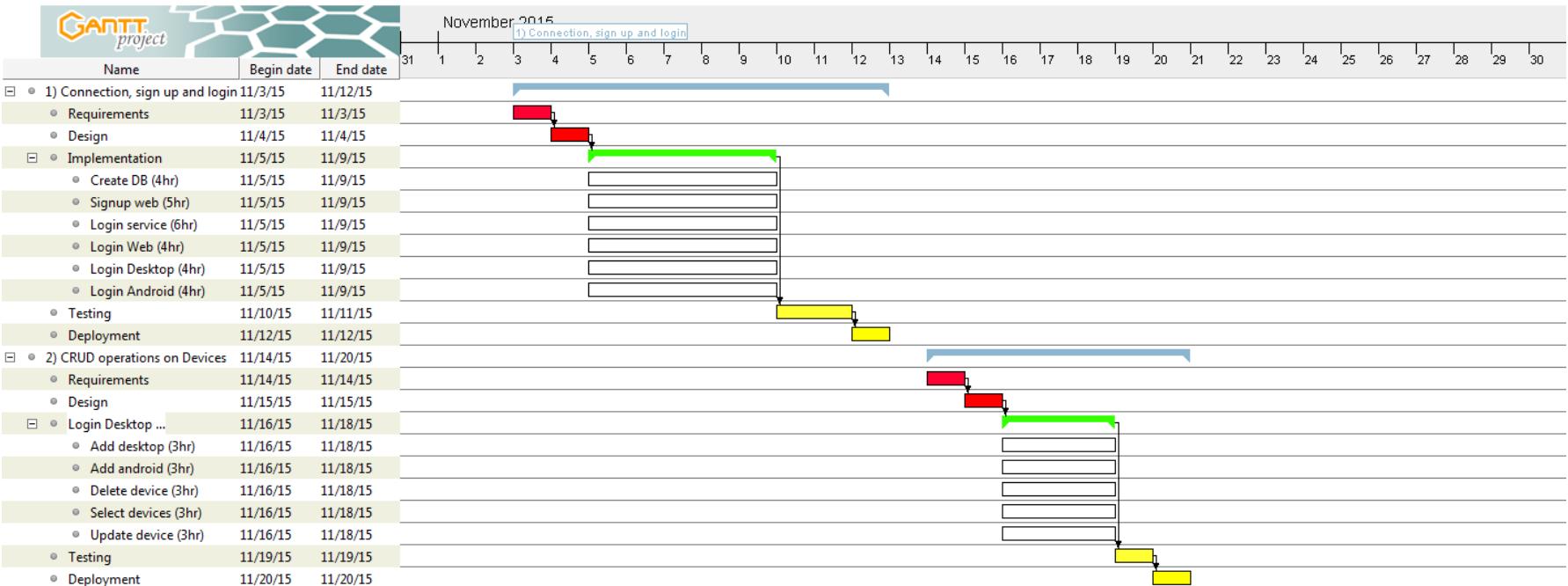


Figure 1:  
Shows Gantt chart sprint 1&2

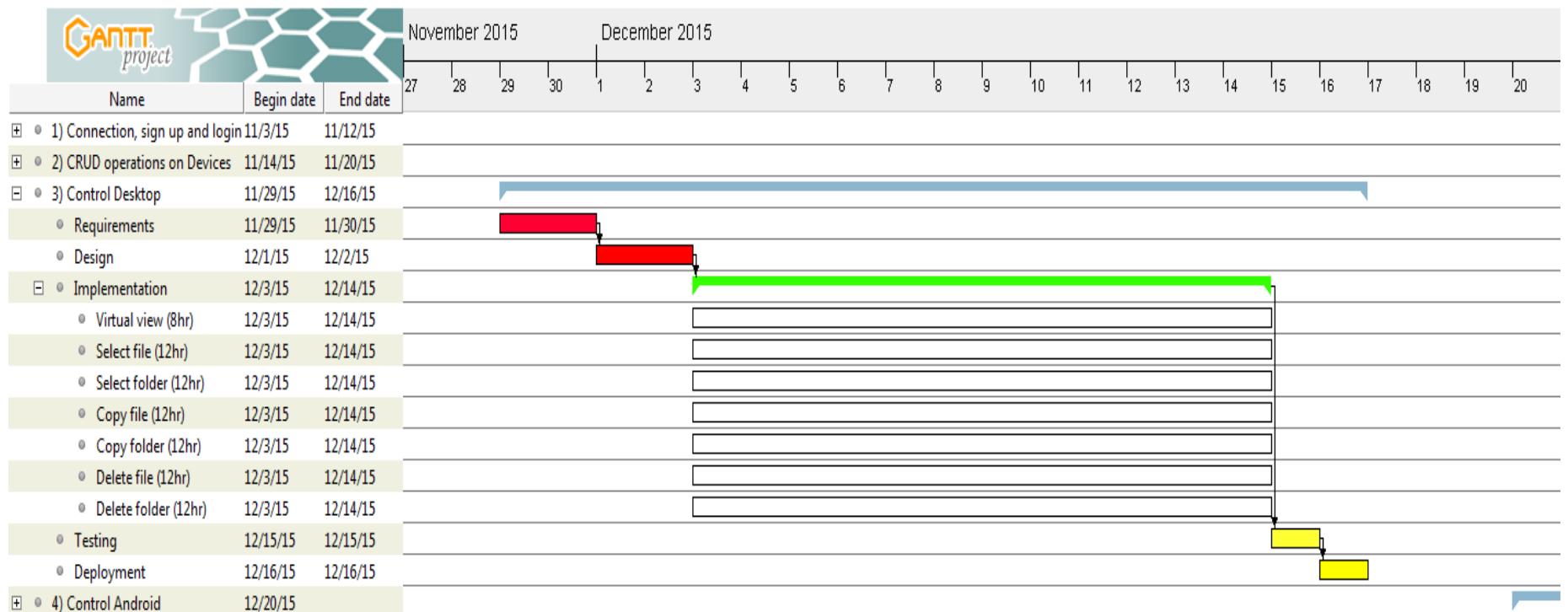


Figure 2:  
Shows Gantt chart sprint 3

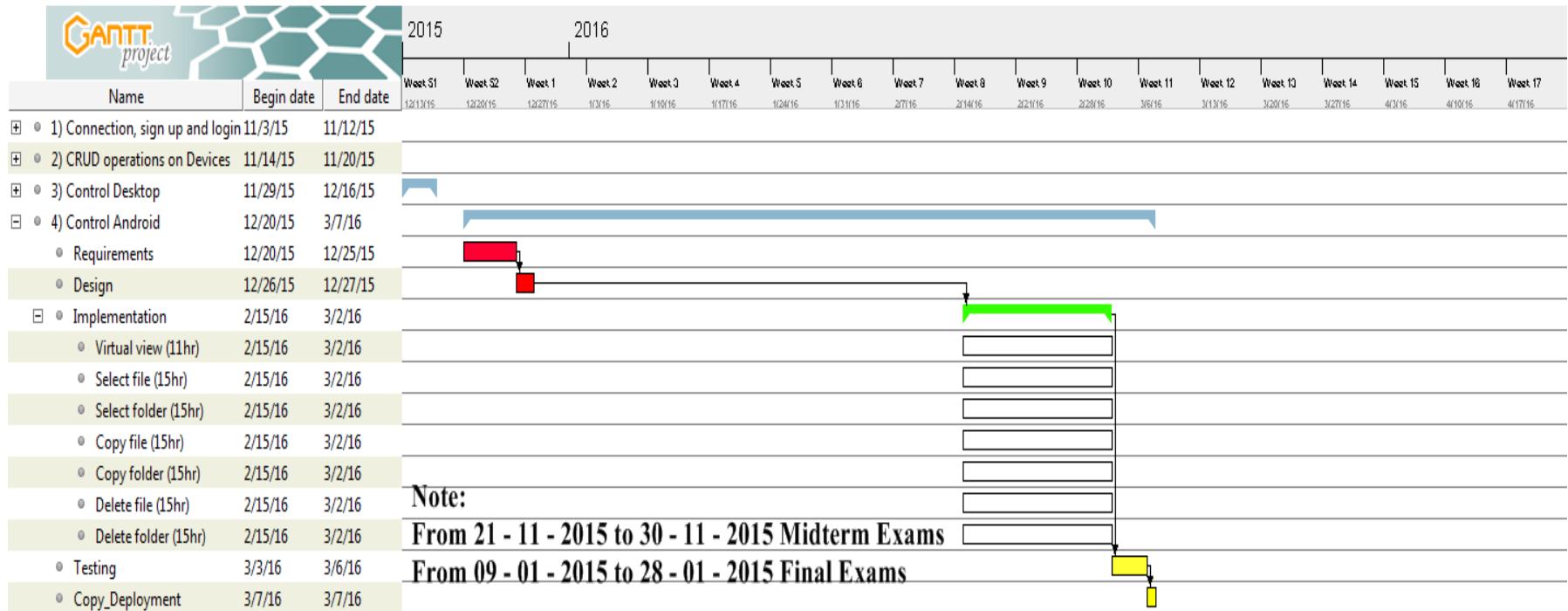


Figure 3:  
Shows Gantt chart sprint 4

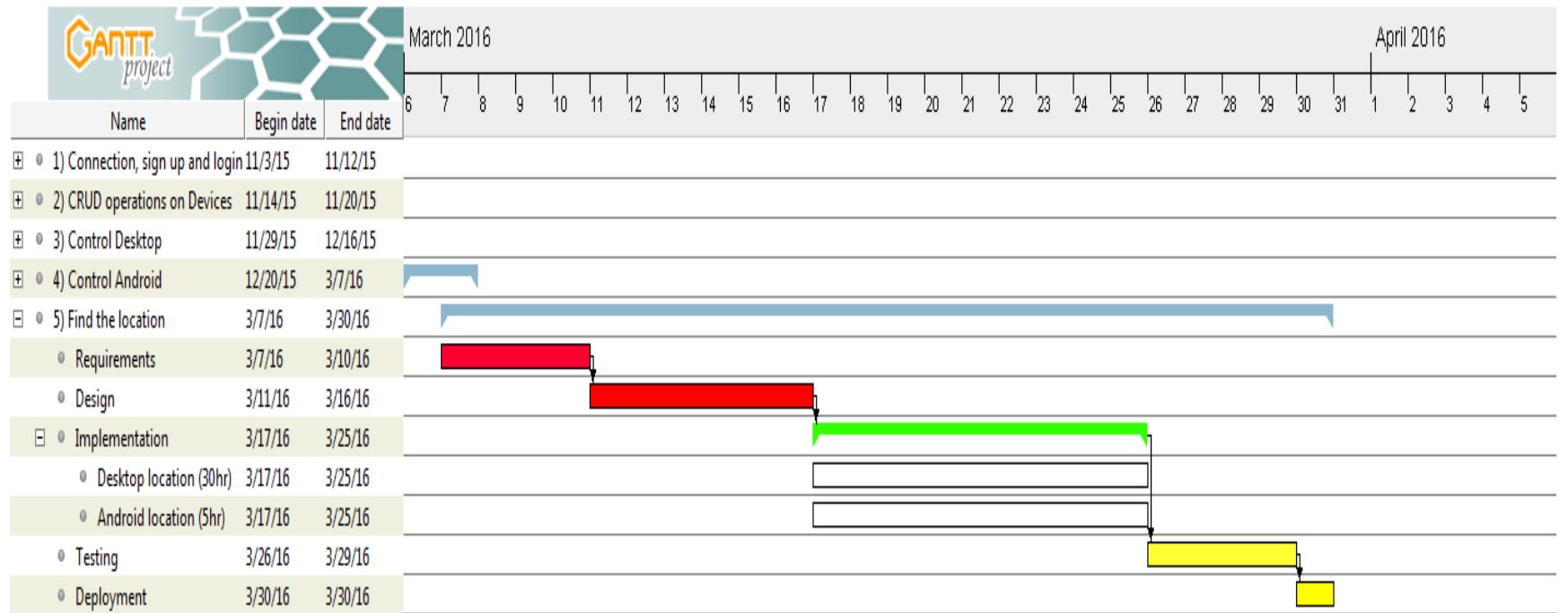


Figure 4:  
Show Gantt chart sprint 5

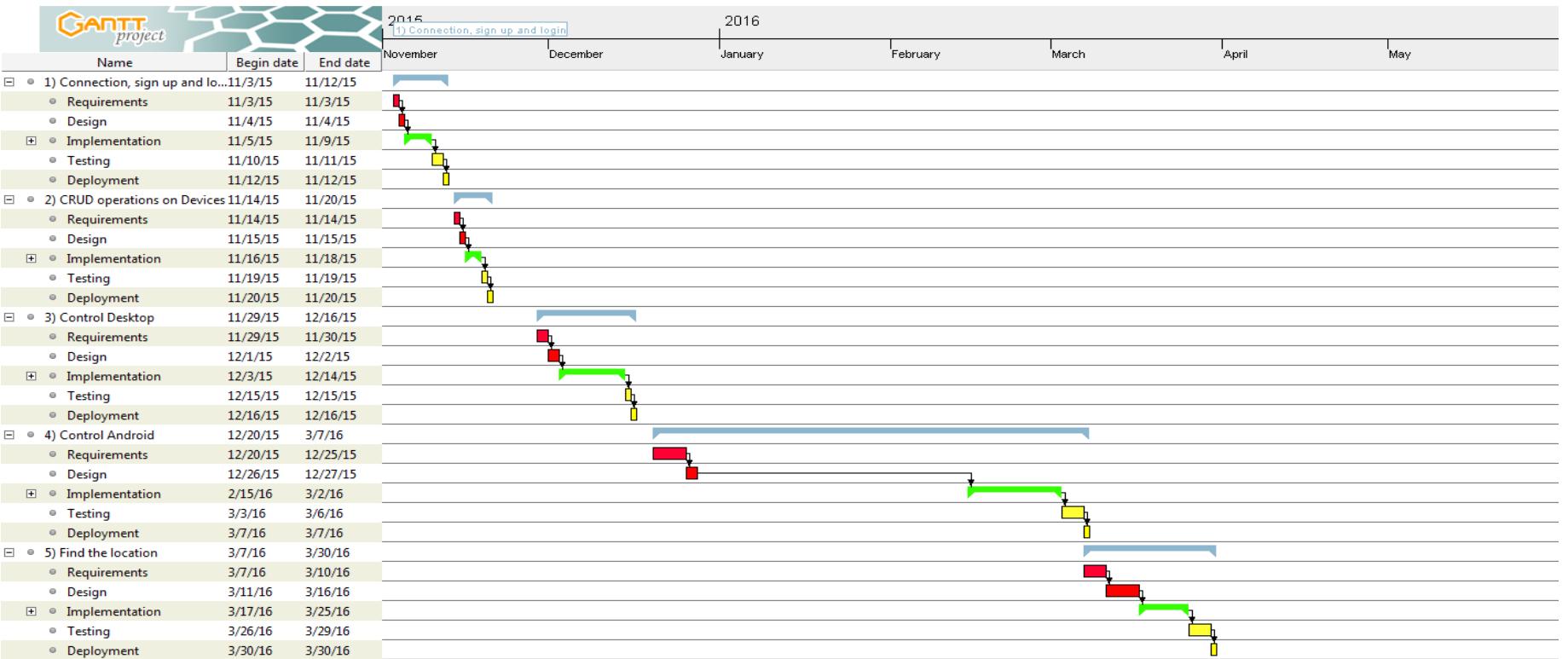


Figure 5:  
Shows the Gantt chart of whole project

# Project development methodology

## We will use agile methodology. Why

The iterative nature of agile development means features are delivered incrementally, enabling some benefits to be released early as the product continues to develop.

Small incremental releases made visible to the product owner and product team through its development help to identify any issues early and make it easier to respond to change. The clear visibility in agile development helps to ensure that any necessary decisions can be taken at the earliest possible opportunity, while there's still time to make a material difference to the outcome.

## How

We will divide our system into 5 sprints each sprint has its requirement phase, Design phase, Implementation phase, Test phase and Deployment phase.

Each sprint has Millstone.

## Millstones

Name	Start Date	End Date	Duration
<b>Establish the connection, sign up and login from web, PC, Android</b>	3 – 11 – 2015	12 – 11 – 2015	8 days
<b>CRUD operations on Devices</b>	22 – 11 – 2015	26 – 11 – 2015	5 days
<b>Control From PC</b>	29 – 11 – 2015	14 – 12 – 2015	16 days
<b>Control From Android</b>	20 – 12 – 2015	07 – 03 – 2015	47 days
<b>Find the Location</b>	07 – 03 – 2015	27 – 03 – 2015	20 days

## Notes

From 14 – 11 – 2015 ☐ ☐ 20 – 11 – 2015

Midterm Exams

From 09 – 01 – 2015 ☐ ☐ 28 – 01 – 2015

Final Exams

## **Sprint 1: Establish the connection, sign up and login**

---

**Start date: 03 – 11 – 2015**

**End date: 12 -11 – 2015**

Name	Start date	End date	Duration
Requirement for sign up from web and login from Web, PC, Android	<b>3 – 11 – 2015</b>	<b>4 – 11 – 2015</b>	<b>1 day</b>
Design the Component of sign up and login	<b>4 – 11 – 2015</b>	<b>5 – 11 – 2015</b>	<b>1 day</b>
Implementation	<b>5 – 11 – 2015</b>	<b>10 – 11 – 2015</b>	<b>5 days</b>
Testing	<b>10 – 11 – 2015</b>	<b>12 – 11 – 2015</b>	<b>2 days</b>
Deploying	<b>12 – 11 – 2015</b>	<b>12 – 11 – 2015</b>	<b>1 day</b>

## **Implementation time plane**

Method Name	Expected Time by hours
Create database	4
Sign up web	5
Login service	6
Login web	4
Login desktop	4
Login android	4

## **Sprint 2: CRUD operations on Devices**

---

**Start date: 22 – 11 – 2015**

**End date: 26 – 11 – 2015**

Name	Start date	End date	Duration
Requirement for CRUD Operations	<b>22 – 11 – 2015</b>	<b>22 – 11 – 2015</b>	<b>1 day</b>
Design the Components of Device	<b>22 – 11 – 2015</b>	<b>22 – 11 – 2015</b>	<b>1 day</b>
Implementation	<b>23 – 11 – 2015</b>	<b>25 – 11 – 2015</b>	<b>3 days</b>
Testing	<b>26 – 11 – 2015</b>	<b>26 – 11 – 2015</b>	<b>1 days</b>
Deploying	<b>26 – 11 – 2015</b>	<b>26 – 11 – 2015</b>	<b>1 day</b>

## **Implementation time plane**

Method Name	Expected time by hours
Add device desktop	3
Add device android	3
Delete device	3
Select devices	3
Update device	3
Delete device	3

## **Sprint 3: Control Desktop**

---

**Start date: 29 – 11 – 2015**

**End date: 14 – 12 – 2015**

Name	Start date	End date	Duration
Requirement	<b>29 – 12 – 2015</b>	<b>1 – 12 – 2015</b>	<b>2 days</b>
Design	<b>1 – 12 – 2015</b>	<b>3 – 12 – 2015</b>	<b>2 day</b>
Implementation	<b>3 – 12 – 2015</b>	<b>13 – 11 – 2015</b>	<b>10 days</b>
Testing	<b>13 – 11 – 2015</b>	<b>14 – 11 – 2015</b>	<b>1 days</b>
Deploying	<b>14 – 11 – 2015</b>	<b>14 – 11 – 2015</b>	<b>1 day</b>

## **Implementation time plane**

Method Name	Expected duration
Virtual view	<b>18</b>
Open folder	<b>12</b>
Copy file	<b>12</b>
Delete file	<b>12</b>
Delete folder	<b>12</b>

## **Sprint 4: Control from android**

---

**Start date: 20 – 12 – 2015**

**End date: 07 – 03 – 2015**

Name	Start date	End date	Duration
Requirement	<b>20 – 12 – 2015</b>	<b>25 – 12 – 2015</b>	<b>5 day</b>
Design	<b>25 – 12 – 2015</b>	<b>27 – 12 – 2015</b>	<b>2 day</b>
Implementation	<b>15 – 02 – 2016</b>	<b>01 – 03 – 2016</b>	<b>15 days</b>
Testing	<b>02 – 03 – 2016</b>	<b>07 – 03 – 2016</b>	<b>6 days</b>
Deploying	<b>07 – 03 – 2016</b>	<b>07 – 03 – 2016</b>	<b>1 day</b>

## **Implementation time plane**

Method Name	Expected duration
Virtual view	<b>11</b>
Open folder	<b>15</b>
Copy file	<b>15</b>
Delete file	<b>15</b>
Delete folder	<b>15</b>

## **Sprint 5: Find the location**

---

**Start date: 07 – 03 – 2016**

**End date: 27 – 03 – 2016**

Name	Start date	End date	Duration
Requirement	<b>07 – 03 – 2016</b>	<b>10 – 03 – 2016</b>	<b>3 day</b>
Design	<b>10 – 03 – 2016</b>	<b>15 – 03 – 2016</b>	<b>5 day</b>
Implementation	<b>15 – 03 – 2016</b>	<b>25 – 03 – 2016</b>	<b>10 days</b>
Testing	<b>26 – 03 – 2016</b>	<b>26 – 03 – 2016</b>	<b>1 days</b>
Deploying	<b>26 – 03 – 2016</b>	<b>27 – 03 – 2016</b>	<b>1 day</b>

## **Implementation time plane**

Method Name	Expected duration
Desktop location	<b>30</b>
Android location	<b>5</b>
Record Voice	<b>6</b>
Record Video	<b>12</b>
Offline Communication	<b>18</b>

## **Technologies:**

---

### **General**

1-Github

### **Front-end**

- 1- JSF
- 2- Java Script
- 3- Jquery
- 4- CSS
- 5- Ajax
- 6- Native android

### **Back-end**

- 1- Java
- 2- Hibernate JPA implementation
- 3- Spring
- 4- Web service
- 5- JPQL
- 6- Mysql
- 7- Socket programming

## **Report Organization:**

---

In chapter two we discuss the closest examples of our project and the main differences between them and our project.

In chapter three we will show analysis of Find My Device, functional and non-functional requirement, use cases and sequence diagrams.

In chapter four the design of Find My Device will be on the spot. We will bring overall class diagram, ERD and software architecture.

In chapter five the summary of project and current state.

# **Chapter 2**

## **Related work**

## **The closest examples of the project and the main differences between them and our project:**

---

According to remote access definition - workers being able to access data or resources from a remote location - there are many closest examples of the project related to remote access which are already done like Team Viewer, Chrome remote desktop, Microsoft remote desktop.

Our project differs from chrome remote desktop and Microsoft remote desktop in supporting file transfer, available as a web app and it's a virtual remote access.

Also it differs from Team viewer in cost, because our project is totally free with no paid versions while Team viewer has commercial version with expensive cost and session time for Team viewer is limited, it breaks down after fixed period of usage.

	<b>Cost</b>	<b>File transfer</b>	<b>Session time</b>	<b>Virtu al</b>	<b>Supported OS</b>	<b>Security</b>
<b>Our project</b>	free	support	Unlimited	Yes	- Available on Windows and the mobile app is available for android - also available <u>as a web app</u> (you need a FMD account)	Yes
<b>Chrome remote desktop</b>	free	Not support	Unlimited	No	Available on Windows, Mac, and Linux. And the mobile app is available for iOS and Android	No
<b>Team viewer</b>	Free but Its paid version is expensive	support	Limited	No	- Available on Windows, Mac, and Linux. And the mobile app is available for iOS, Android, Windows Phone, and BlackBerry - also available <u>as a web app</u> (you need a Team Viewer account)	No
<b>Microsoft remote desktop</b>	free	Not support	Unlimited	No	Available on WindowsAnd the mobile app is available for iOS, Windows phone and Android	No

# Chapter 3

## System Analysis and Design

## Sprint 1 – Establish connection

### Sprint specification:

---

#### Functional requirement:

1. User login

A user must login to interact with the system

2. User sign-up

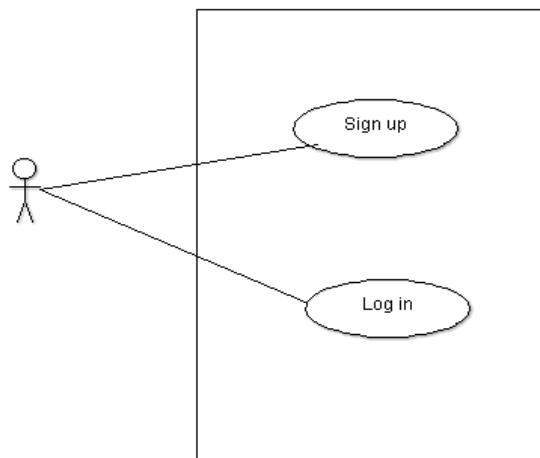
A user must register to create account on the system.

#### Non-Functional requirement:

- **Response time:** 3s
- **Throughput:** 20 operation/min
- **Availability:** 5 Running days
- **Allowances for reusability:** 75%

### Use case diagram:

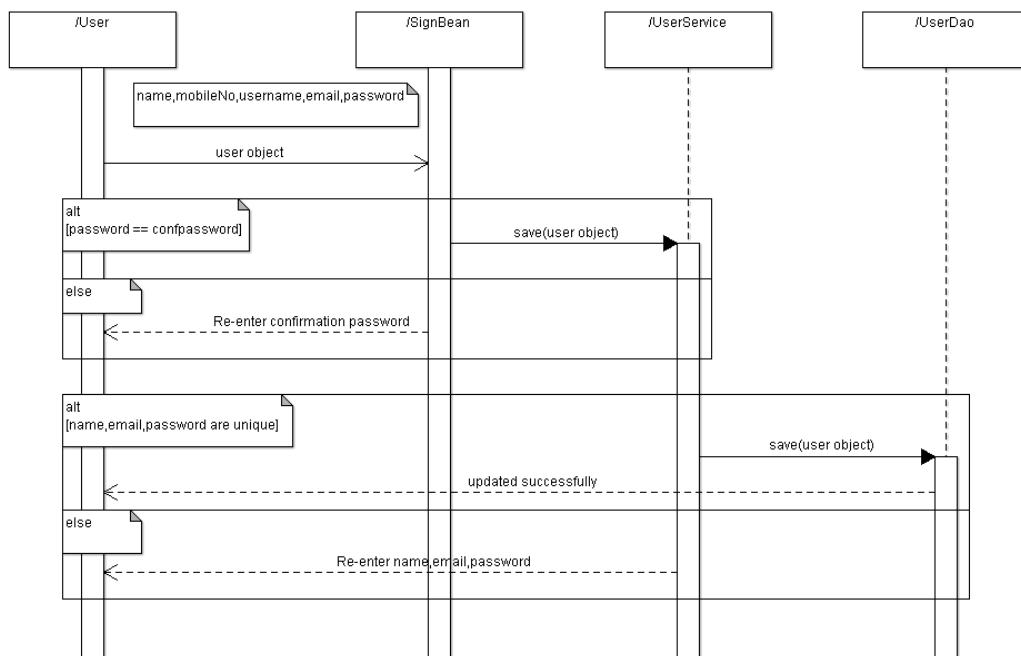
---



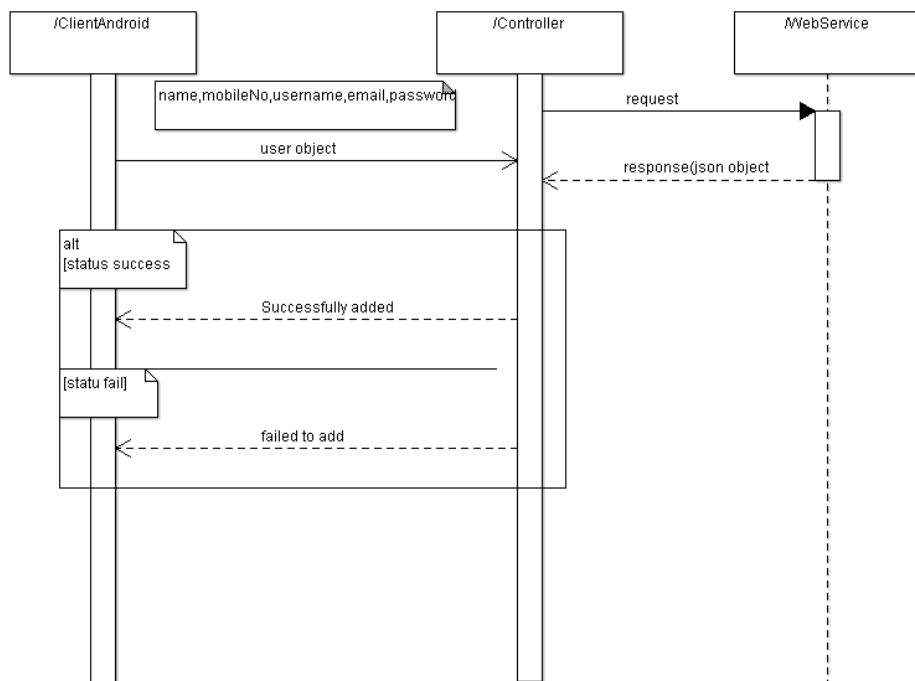
## **Sequence diagrams:**

---

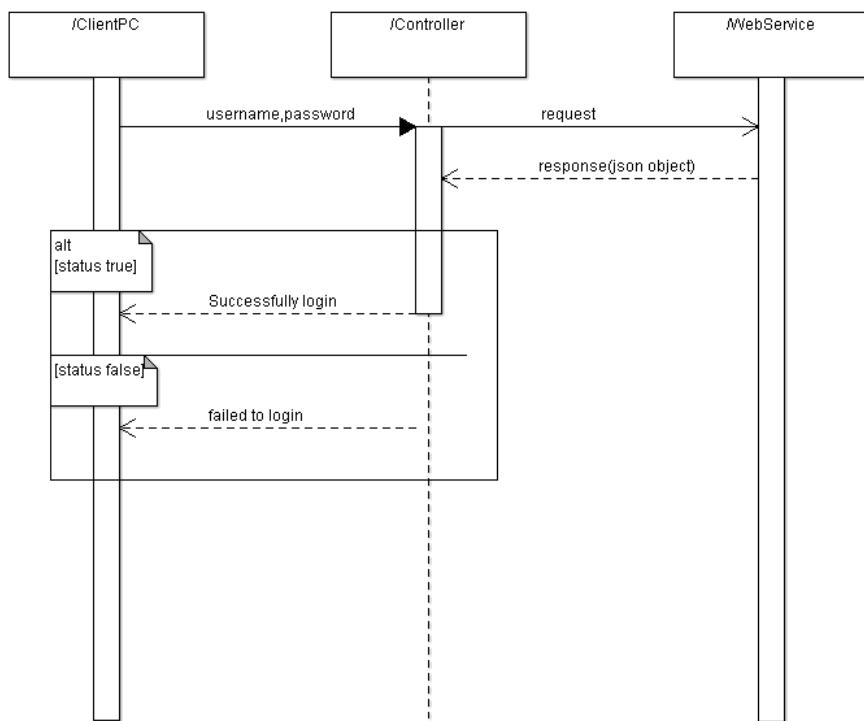
### **Sign up from web**



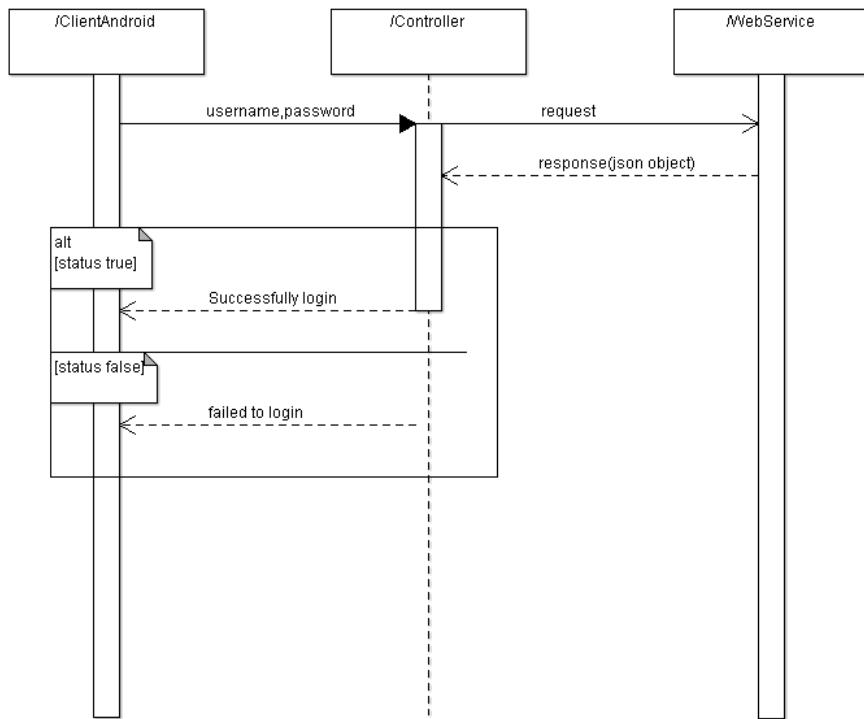
## Sign in from android



## Log in from desktop



## Log in from android



## **Sprint 2 –CRUD operations on Device**

### **Sprint specification:**

---

#### **Functional requirement:**

1- Add device

Access from desktop or android, user must be logged in

The device added once

2- View device

Access from website only, user must be logged in

3- Update device

Access from website only, user must be logged in and must choose device

4- Delete device

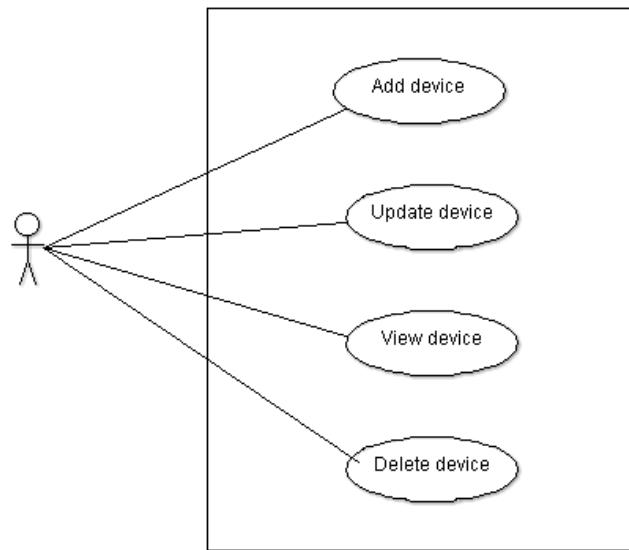
Access from website only, user must be logged in

#### **Non-Functional requirement:**

- **Response time:** 4s
  
- **Throughput:** 15 operation/min
  
- **Availability:** 5 Running days
  
- **Allowances for reusability:** 42%

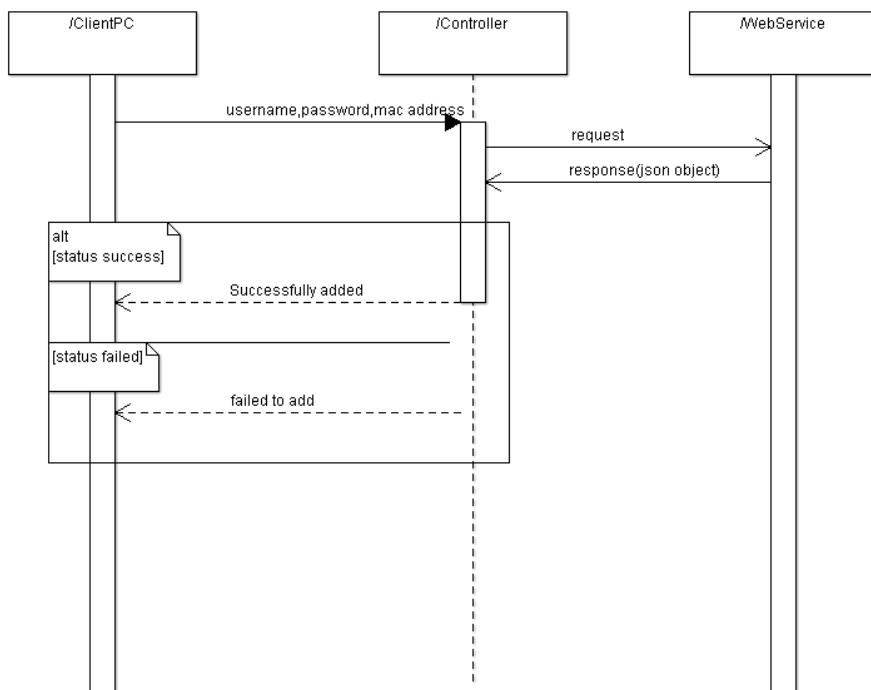
## Use case Diagram:

---

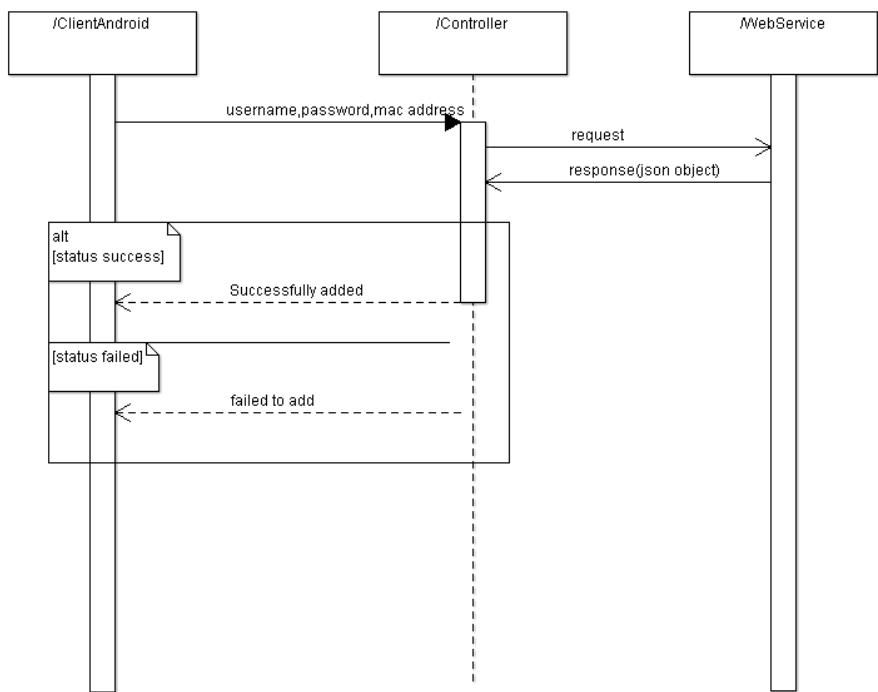


## Sequence Diagrams

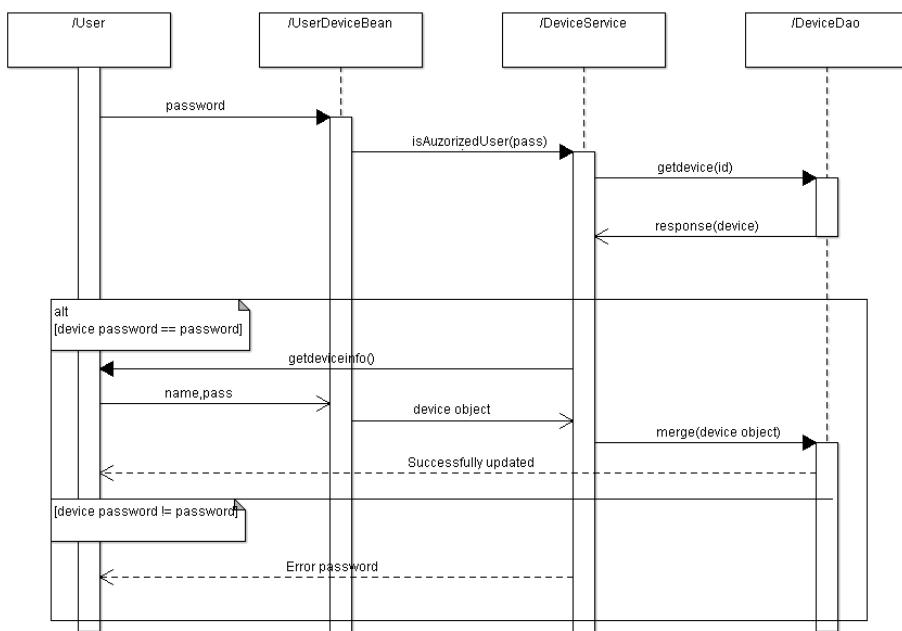
### Add device from desktop



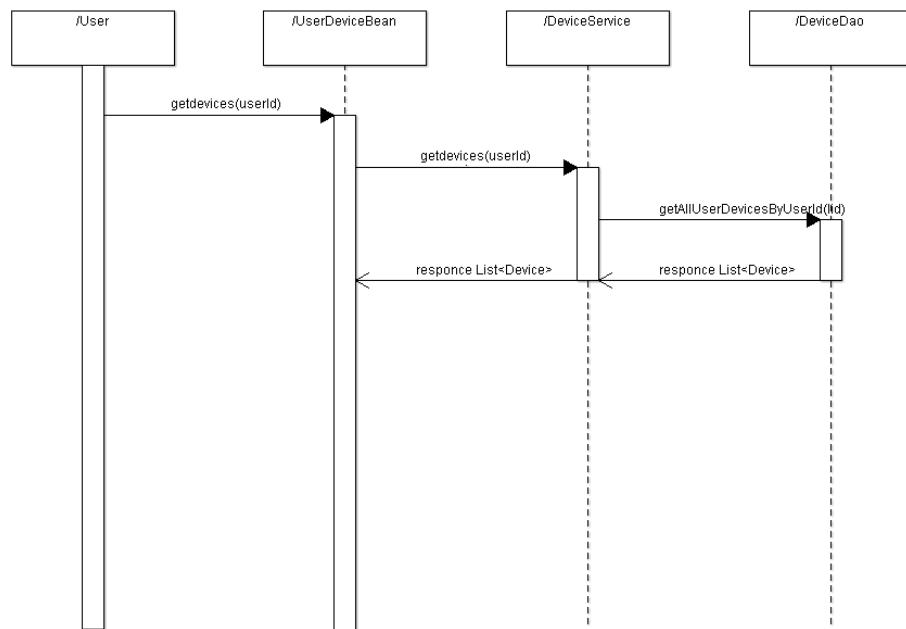
## Add device from android



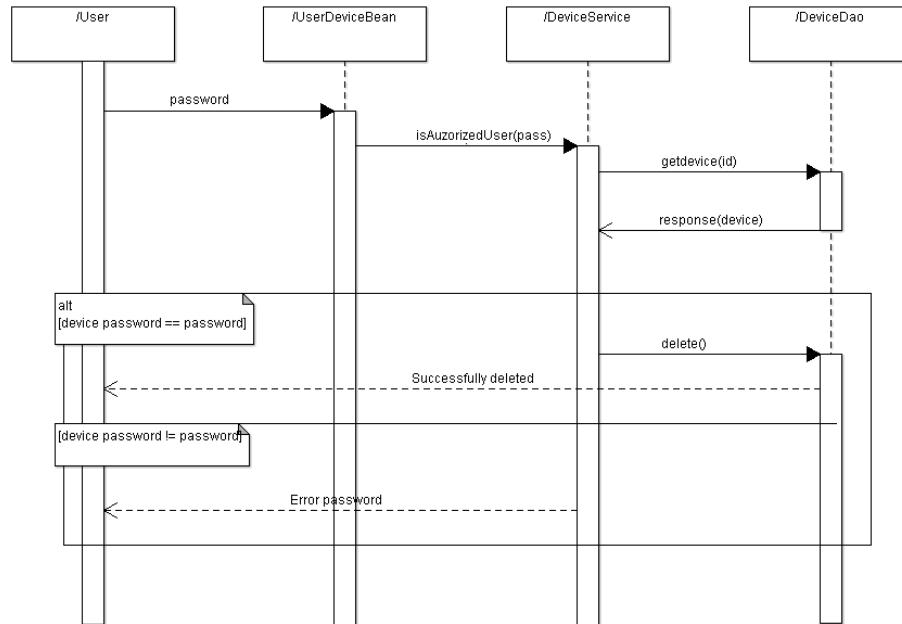
## Update device from web



## View device from web



## Delete device from web



## **Sprint 3 & 4 – Communicate**

### **Sprint specification:**

---

#### **Functional requirement:**

1- Connect to device

Open a live connection between clients and sever

2- Send and receive command

Send commands from web to client and receive response to update view of user

3- Retrieve file

Retrieve files from client machine to server

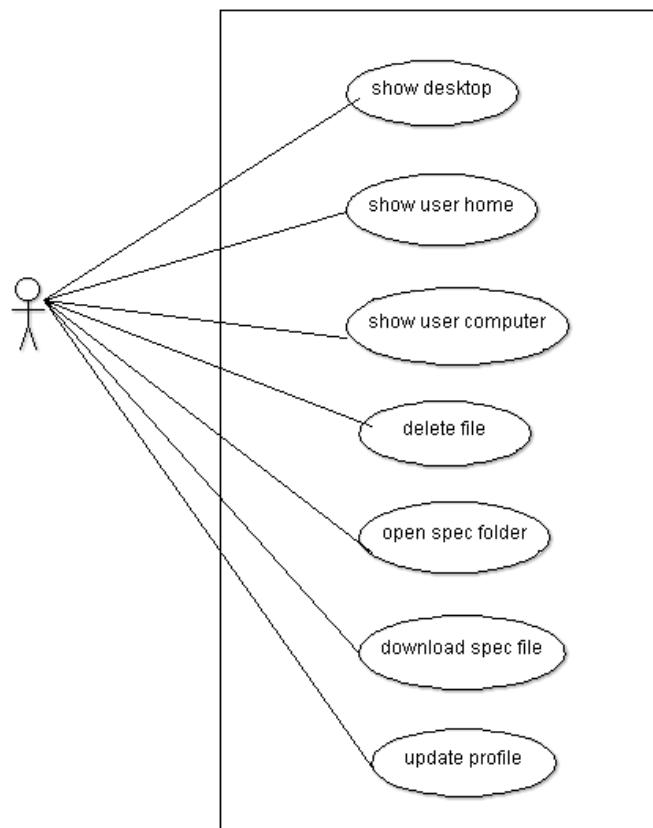
#### **Non-Functional requirement:**

- **Response time:** 6s
- **Throughput:** 10 operation/min
- **Allowances for reusability:** 15%

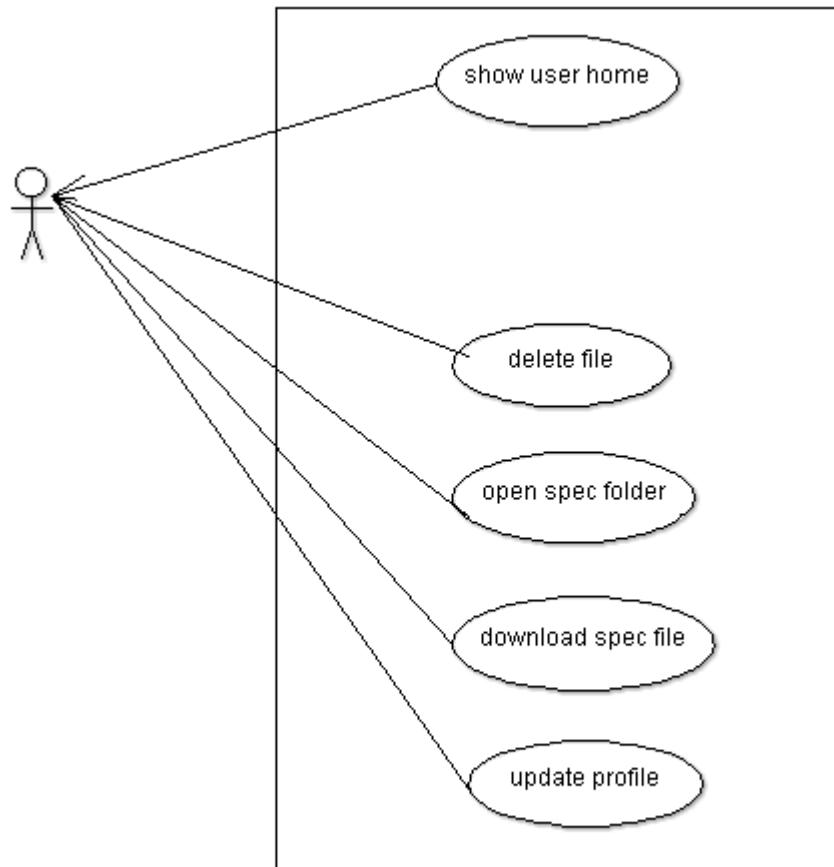
## Use case Diagram:

---

- Sprint 3



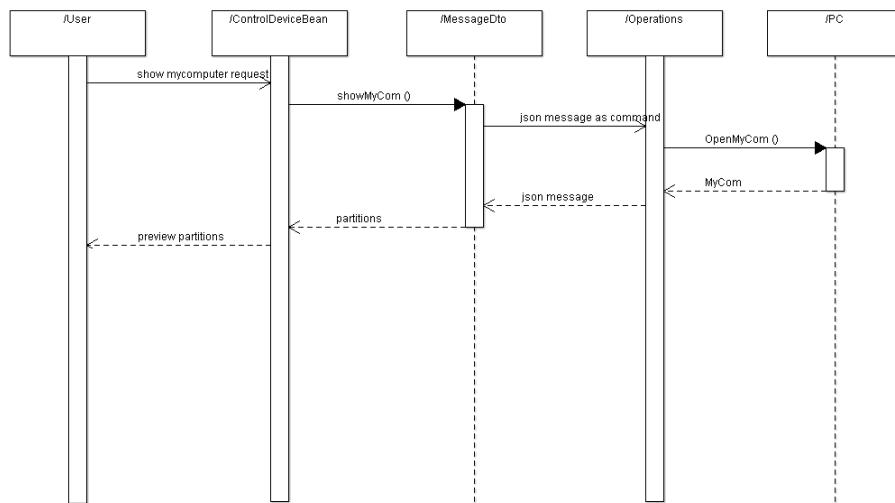
- Sprint 4



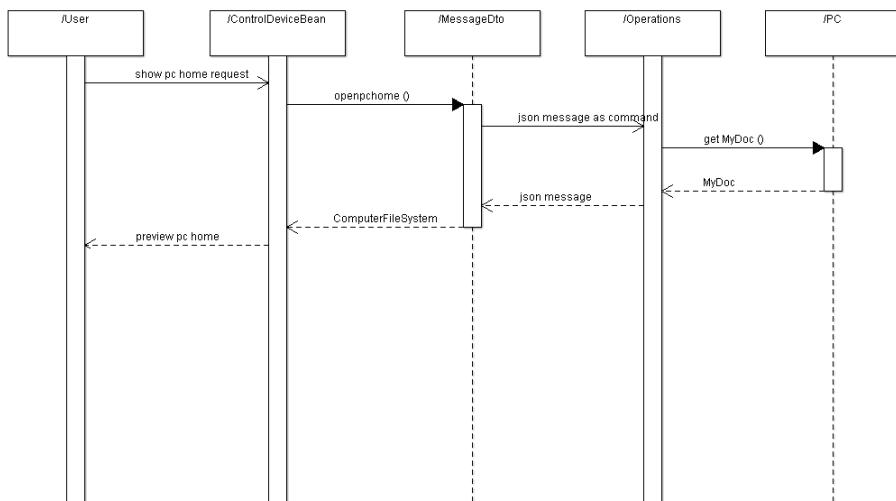
## Sequence Diagrams:

---

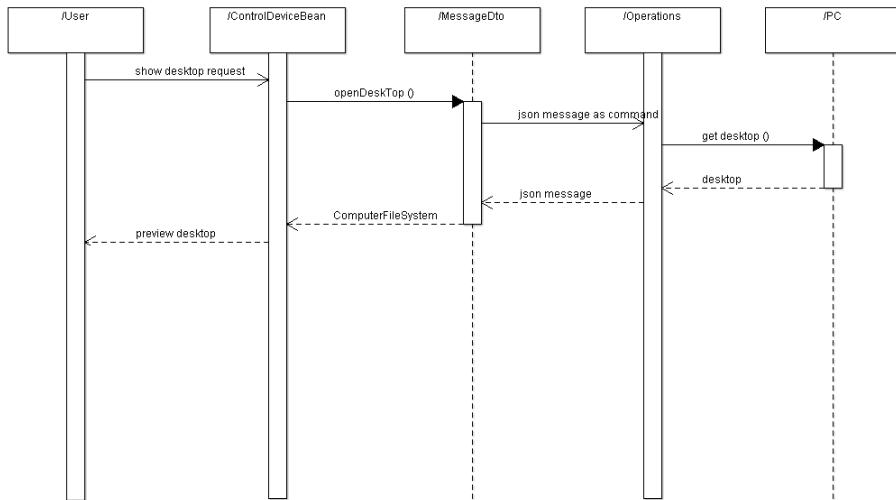
### Show user computer (Sprint 3)



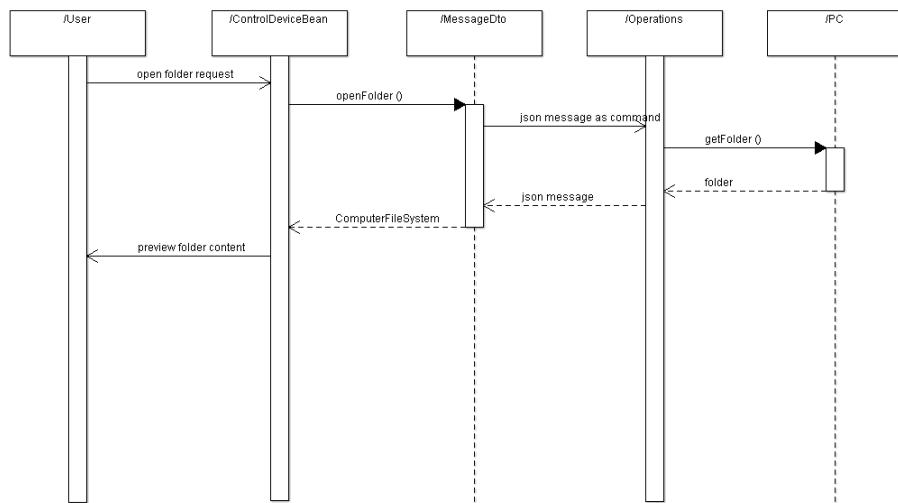
## Show user home (Sprint 3 & 4)



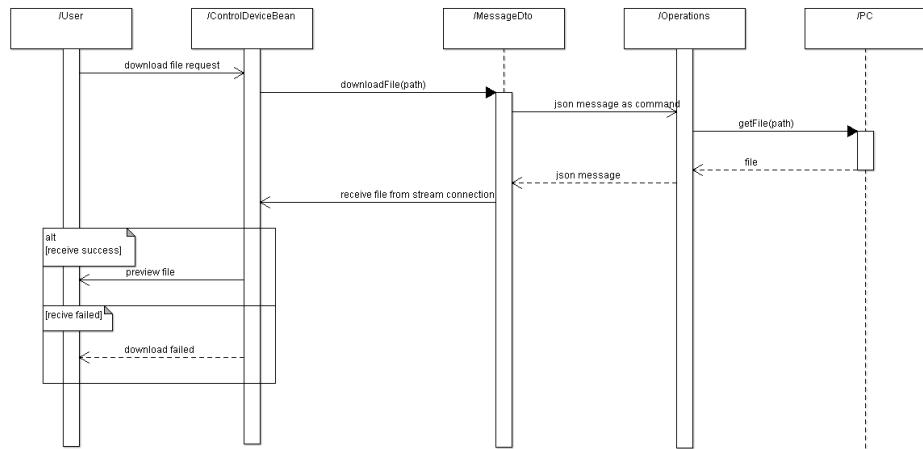
## Show desktop (Sprint 3)



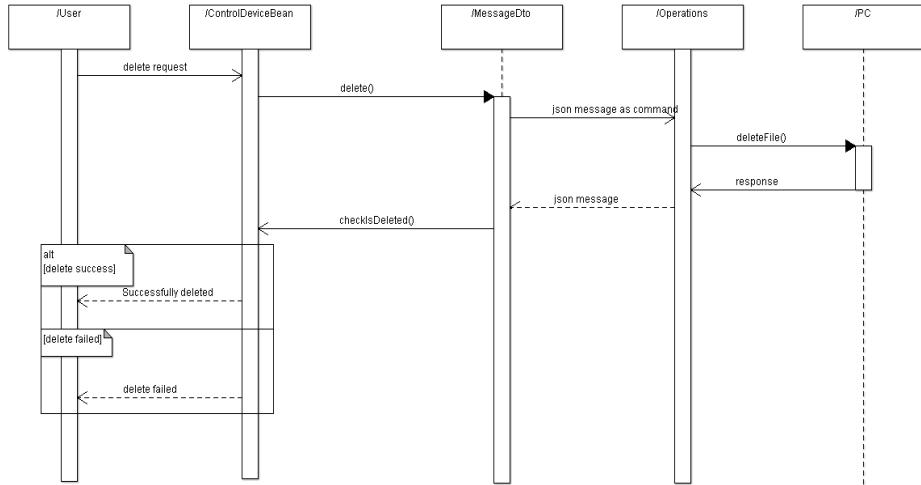
## Open specific folder (Sprint 3 & 4)



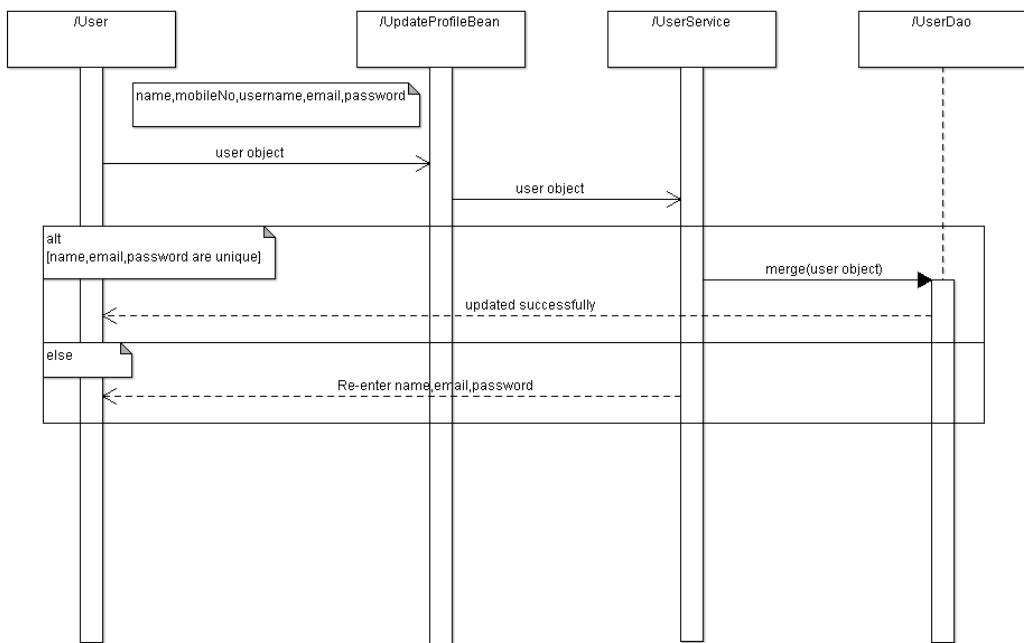
## Download specific file (Sprint 3 & 4)



## Delete file (Sprint 3 & 4)



## Update user profile (Sprint 3)



## **Sprint 5 – Find device location**

### **Sprint specification:**

---

#### **Functional requirement:**

##### 1. Device location

A user can request to find device location

##### 2. Record sound

A user can request to record sound

##### 3. Record video

A user can request to record video

##### 4. Offline communication

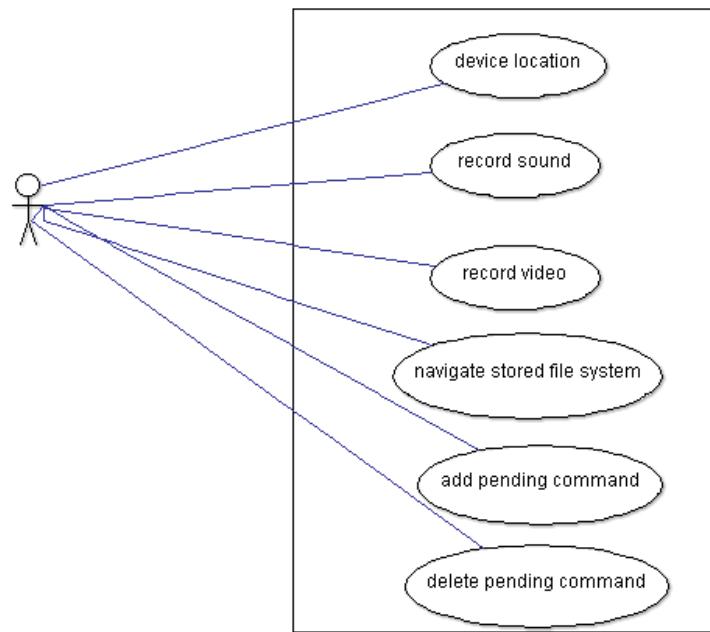
A user can navigate stored file system image, add pending command, and delete pending command

#### **Non-Functional requirement:**

- **Response time:** 1s
- **Throughput:** 60 operation/min
- **Availability:** 7 Running days
- **Allowances for reusability:** 35%

## Use case diagram:

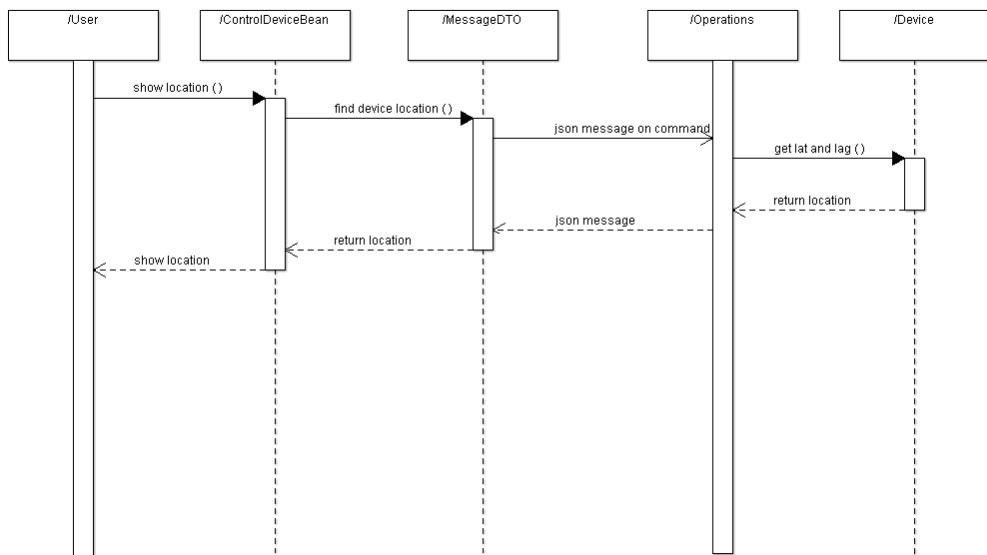
---



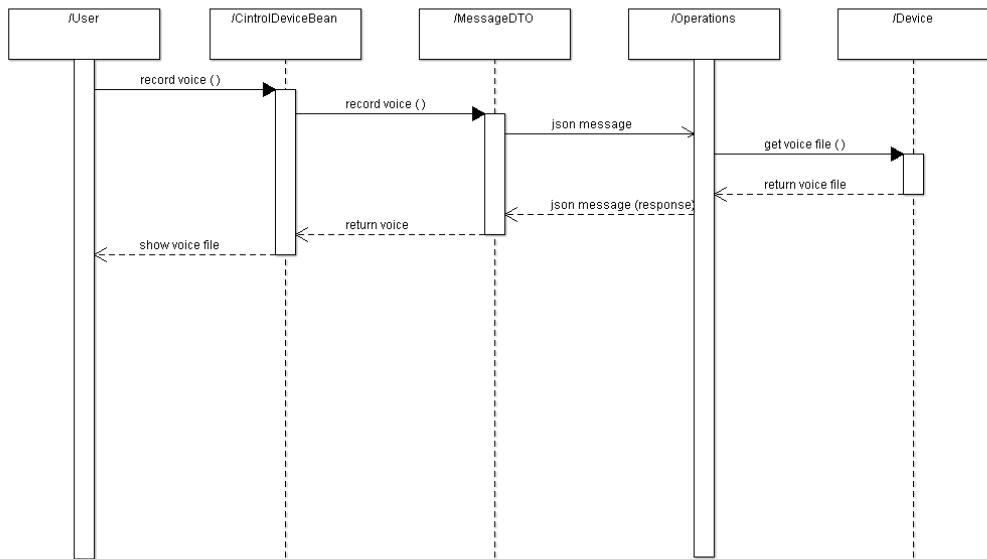
## **Sequence Diagrams:**

---

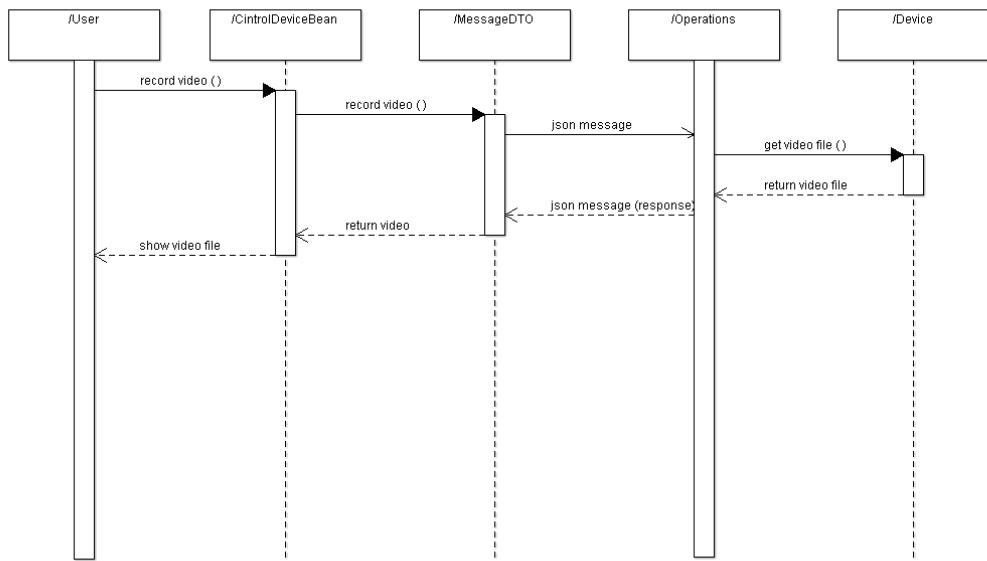
### **Device location**



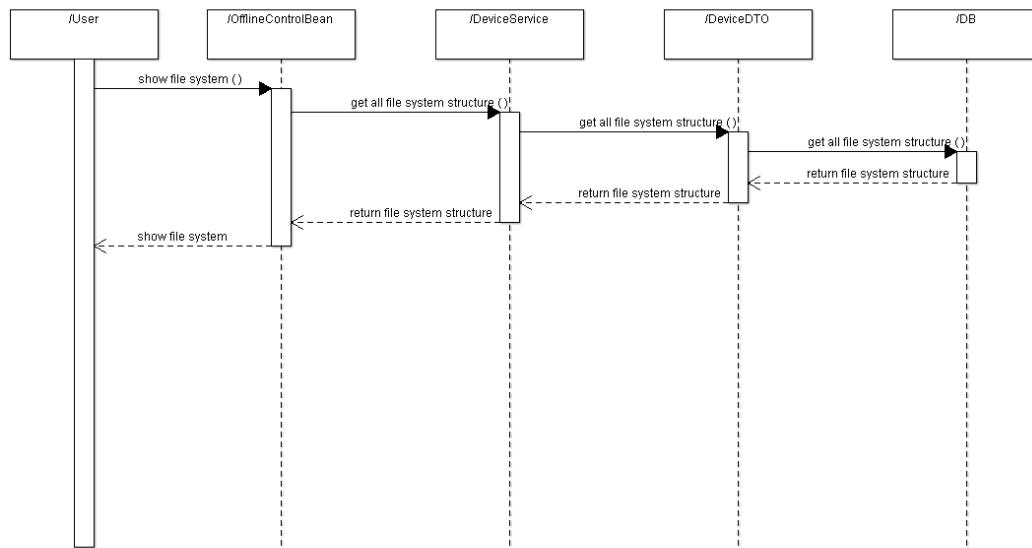
## Record voice



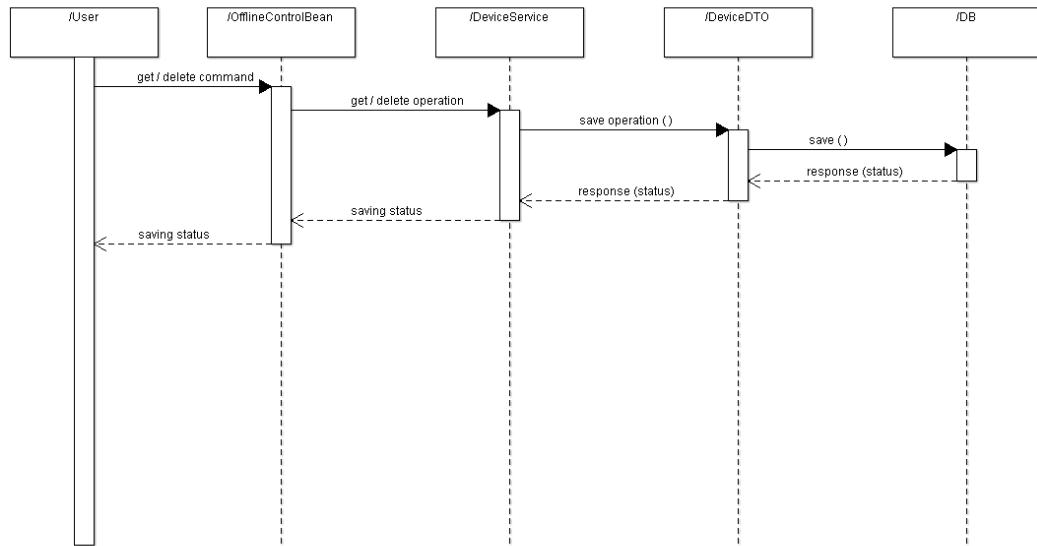
## Record video



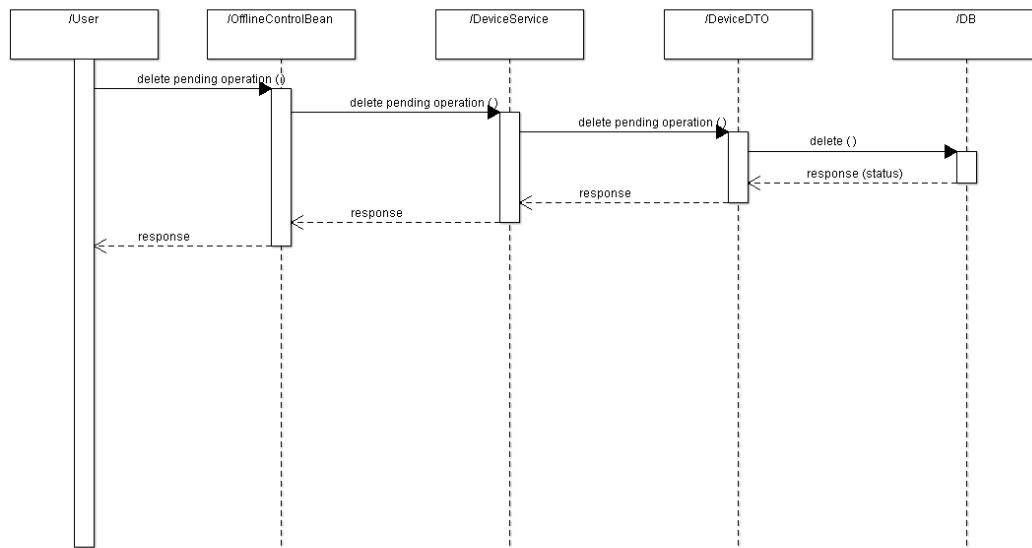
## Navigate stored file system



## Add pending command



## Delete pending command



## **Testing Phase**

We use some automated tools like Junit in java to automate the testing of some methods that can be tested automatically and other system functionality tested manually by us.

**Firstly we did unit testing and then integration test of various component of the system.**

We divide the testing phase to 3 phases (one per each application).

### **Desktop PC Application:**

We have testing module package which contains 4 classes.

1. File System Test class:
2. This class tests the file system navigation methods and checks the behavior of retrieving the content of spec folder, desktop or partitions info on various operating systems like Windows and Linux.
3. Test deleting file or folders from the hard disk.
  
4. Location Test class:
  - a. This class tests the method of retrieving the location of the pc.
  
5. Recording Test class:
  - a. This class tests the recording operation on pc and generate output file of recording.
  
6. Utility Test class:
  - a. This class tests the behavior of various utility methods like test getting the mac address of the current device ... etc.

The other methods tested manually via running the server and installing the application on the pc and check the behavior of the application with different test cases.

## **Android Application:**

All android methods tested manually via running the server and installing the application on the mobile and check the behavior of the application with different test cases.

Testing includes:

1. Sign up new user and login (test success, failure and null scenarios).
2. Register the device to the server.
3. Logout from the application.

## **Web Application:**

We have testing module package which contains.

1. User Web Service Test class:
2. This class tests new user sign up service with 2 possible scenario success and failure.
3. Test user login service by email or username with 2 possible scenario success and failure.
4. Location Web Service Test class:
  - a. This class tests the method of retrieving adding location of spec pc or mobile to the server database.
5. Device Web Service Test class:
  - a. This class tests new registered device service with 2 possible scenario success and failure.

The other methods tested manually via:

1. Running the server and creating a new user and login with that user.
2. Installing client application on pc or smart phone and register the device to the system.
3. Access this device through the web application and check the file system navigation and checking retrieving location , recording voice , recording video , change configuration of recording response time ...etc.
4. Test deleting device and change user information's.

## **Whole System Test:**

### **At the end of the project:**

1. We made system and performance test to check the response time and throughput of the system.
2. Made usability test by some real user to check the ease of which is the application and how functionalities looks like

The results of this test is that we change the UI and adding photos, icons, themes, template to make the system more easy to use by the users without any difficulties.

## **Suggested solution:**

---

Developing "Find my device" system which mange you to find your device via web browser and pre-installed client on device in case of stolen or lost. Simply you install simple client on your device (desktop and android) and register this device then can controlled later

**Primarily design and architecture:**

---

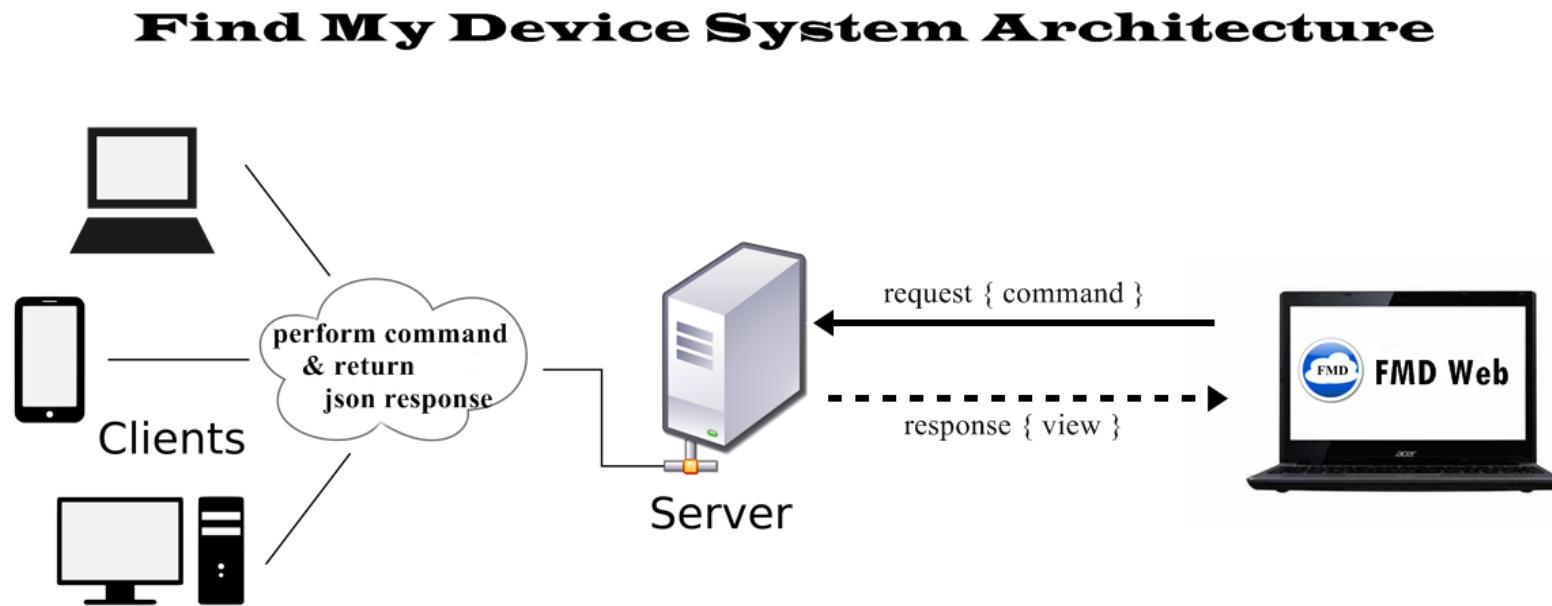


Figure 6:  
Shows system architecture

## Entity Relationship Diagram:

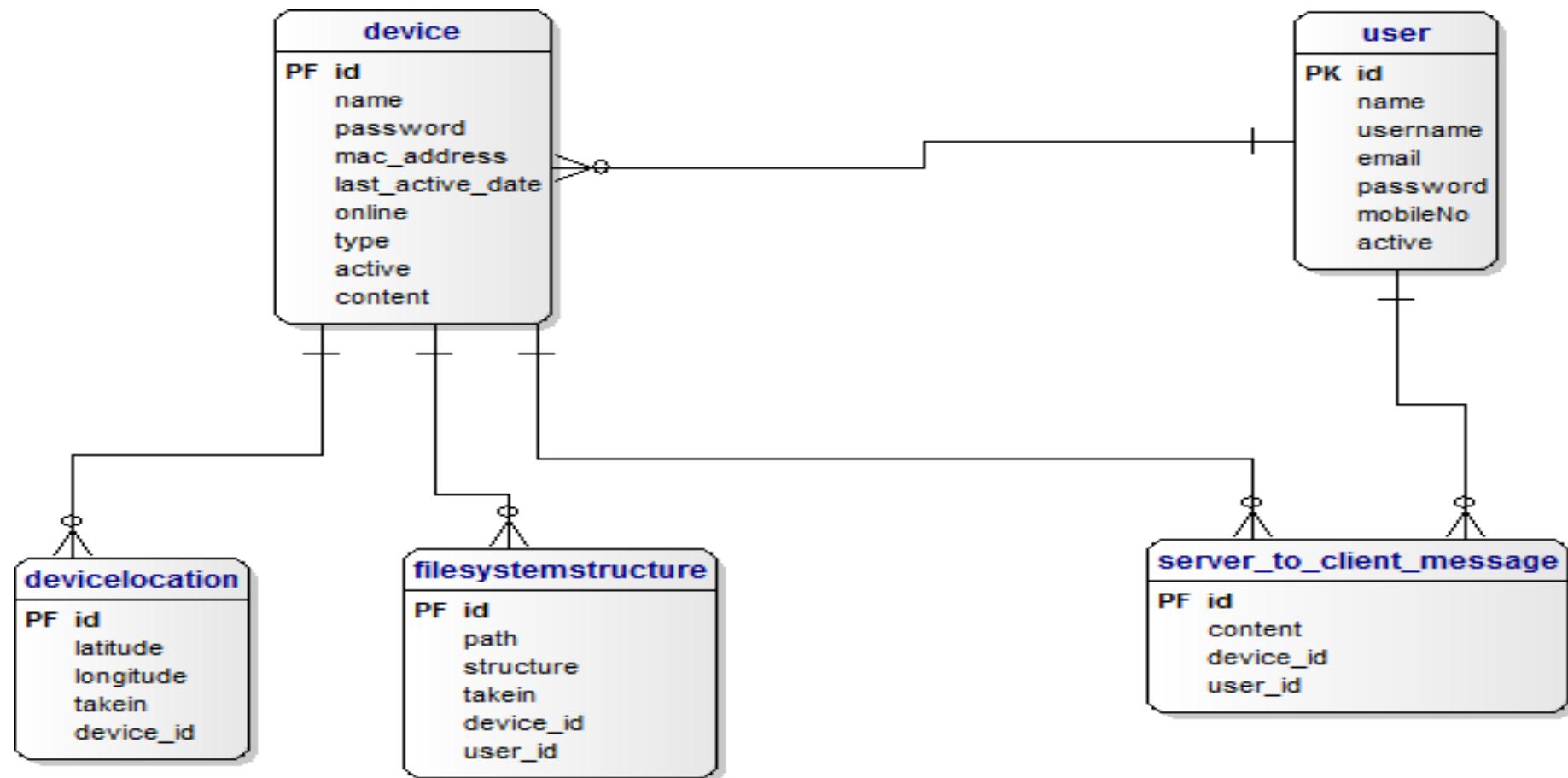
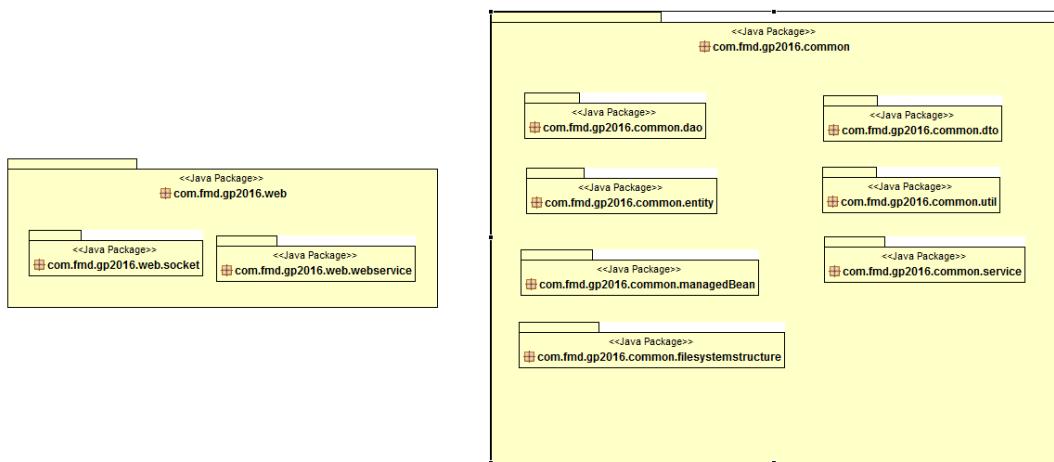


Figure 7:

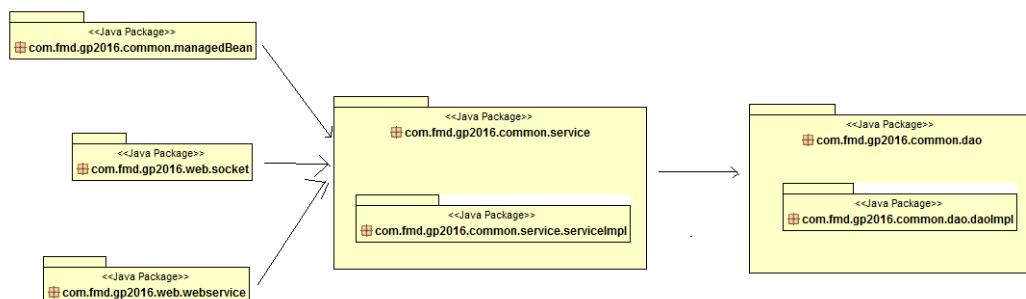
Shows system ERD

# Class Diagram for Server Software

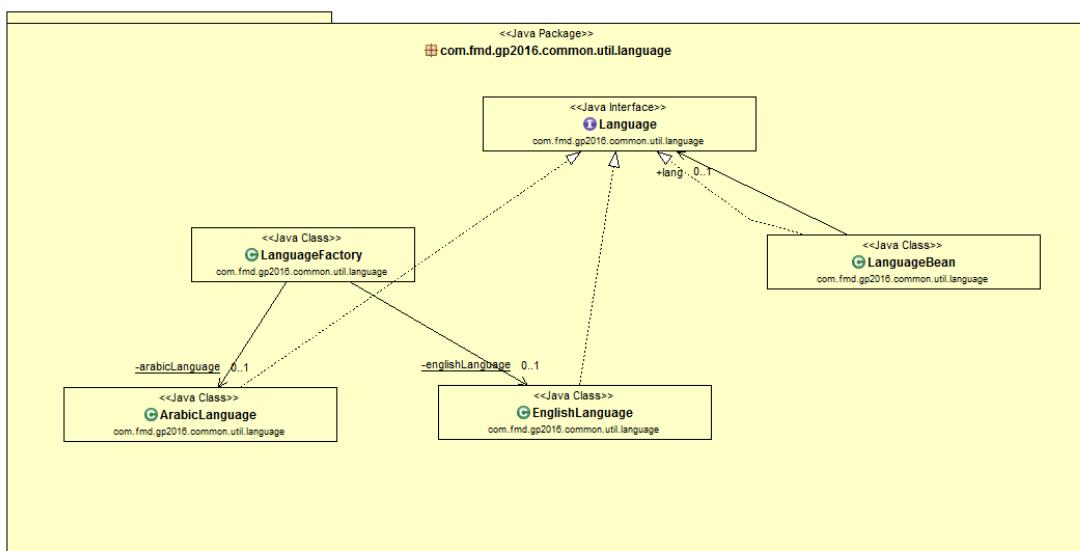
## Whole System View



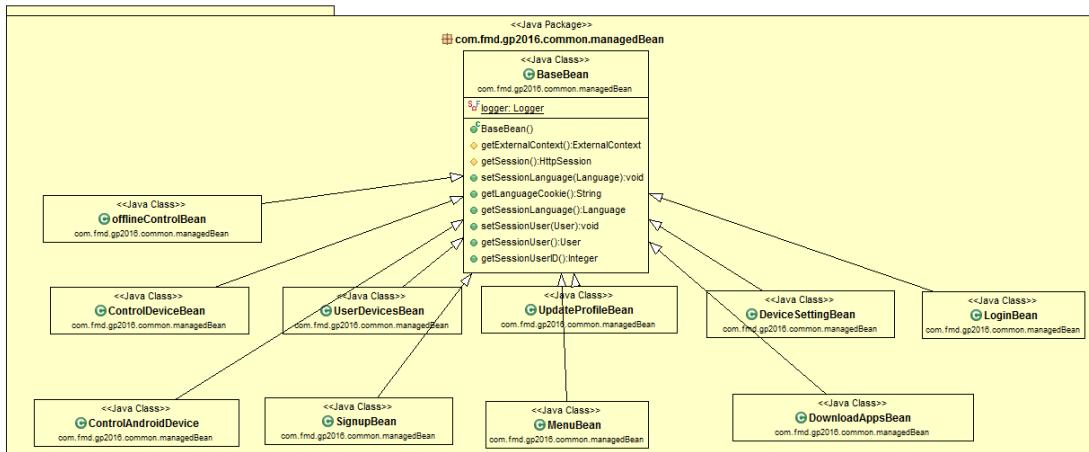
## database Access Structure



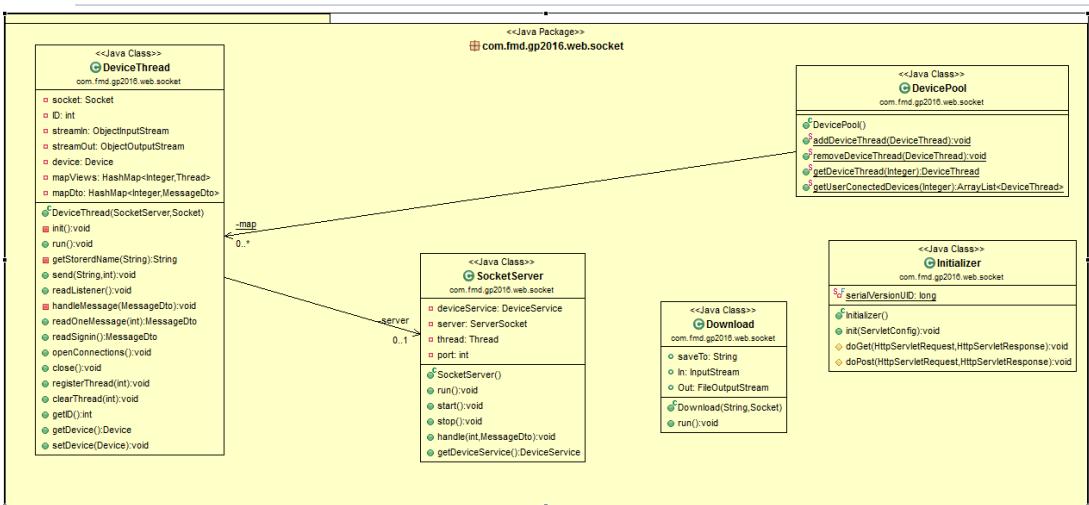
## Localization support



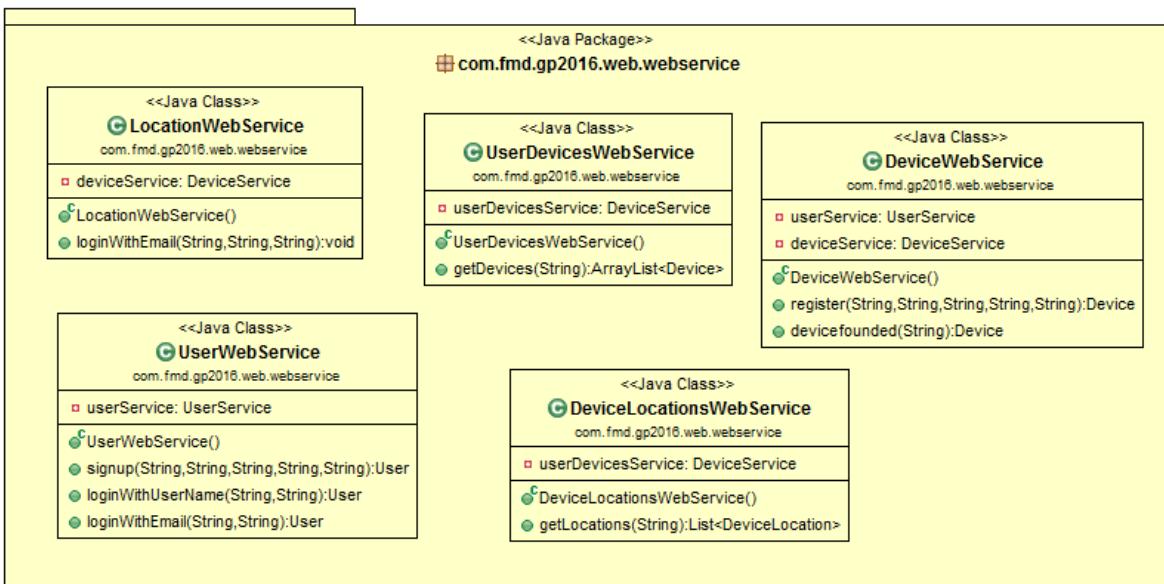
## View Structure



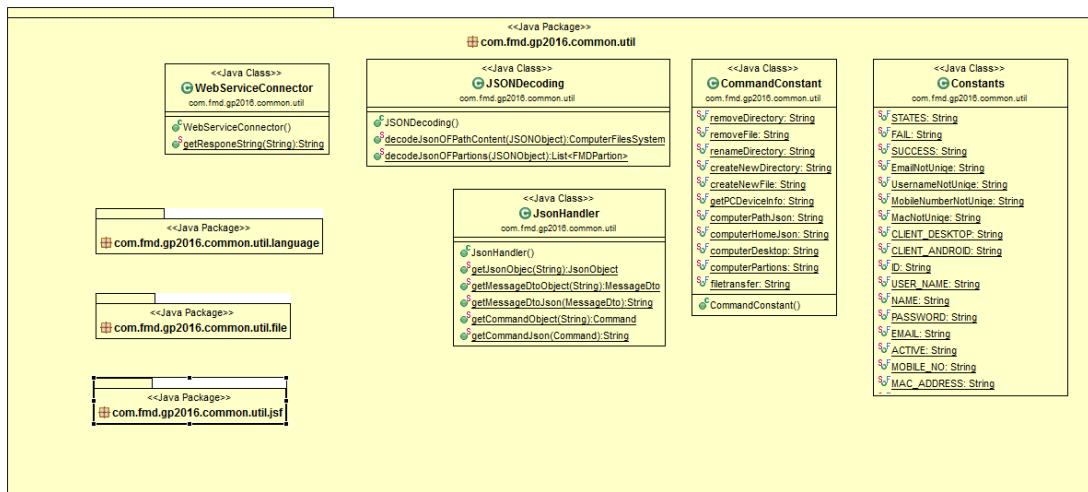
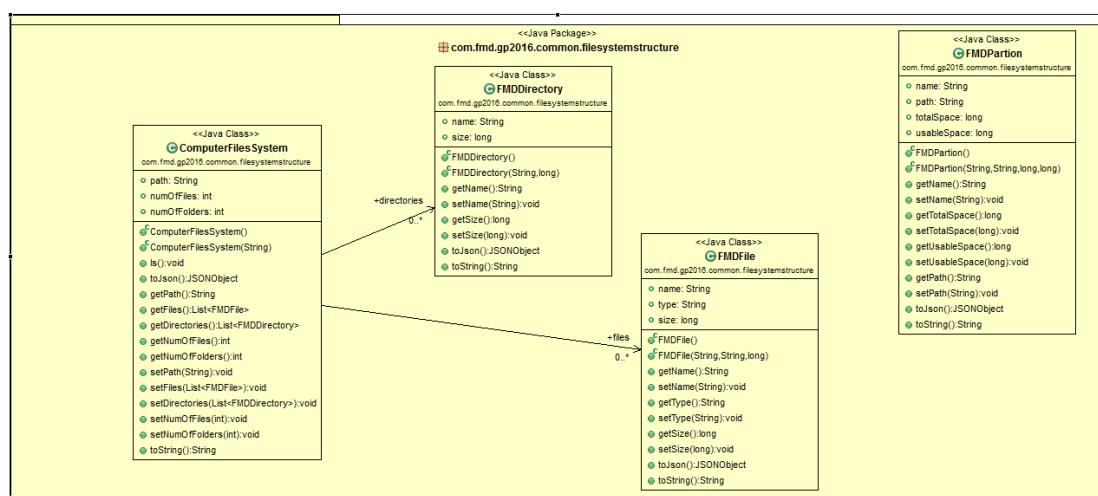
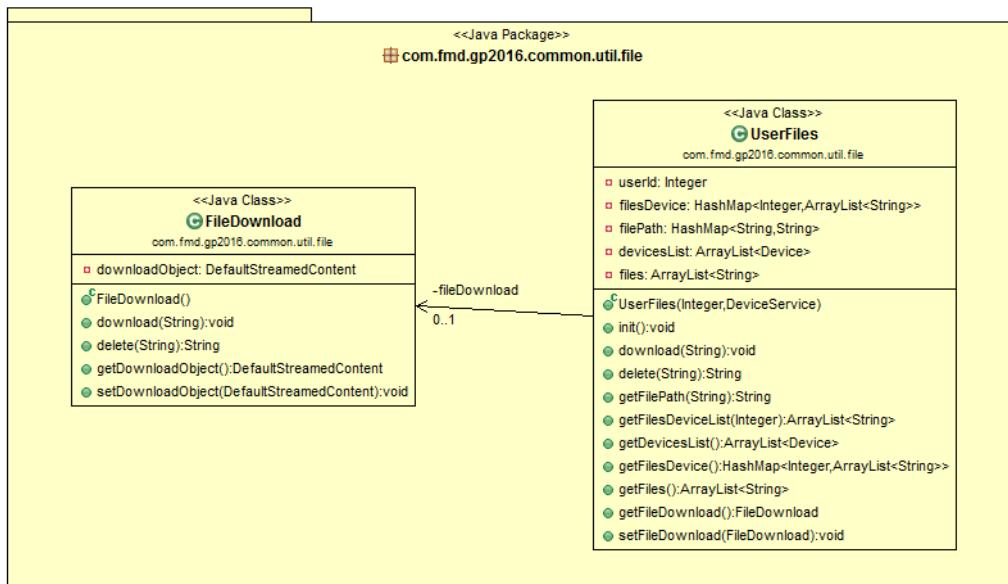
# Socket Programming package

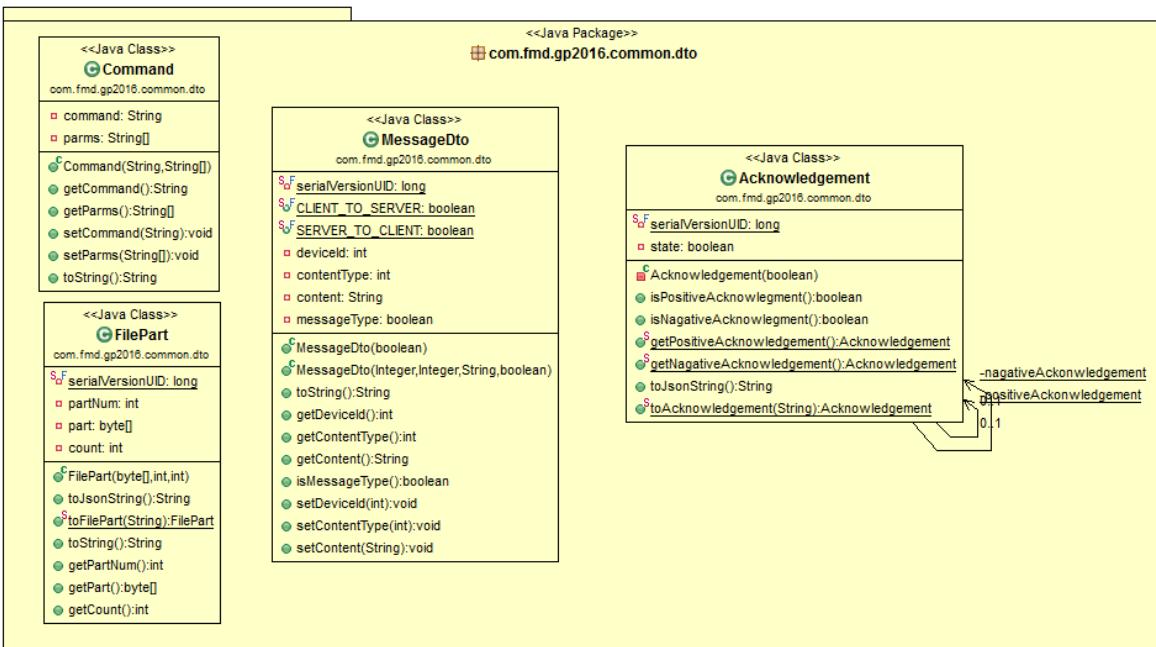


## Web services



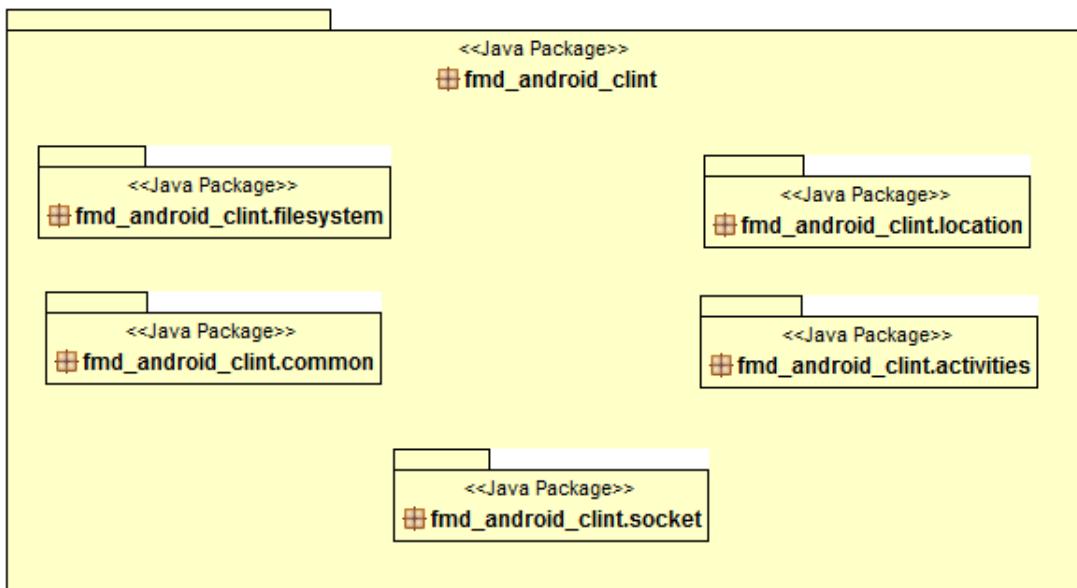
## The following utilites and helper methods



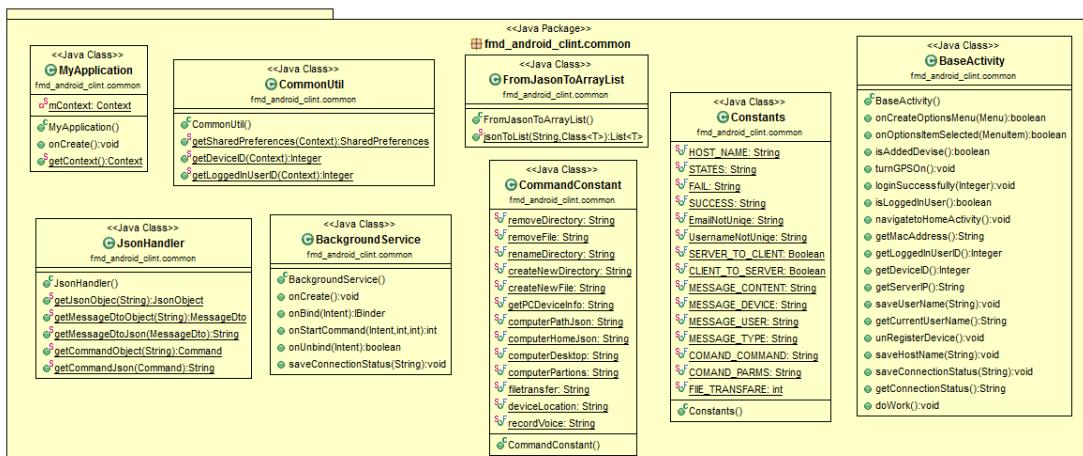


# Class Diagram for Android Application

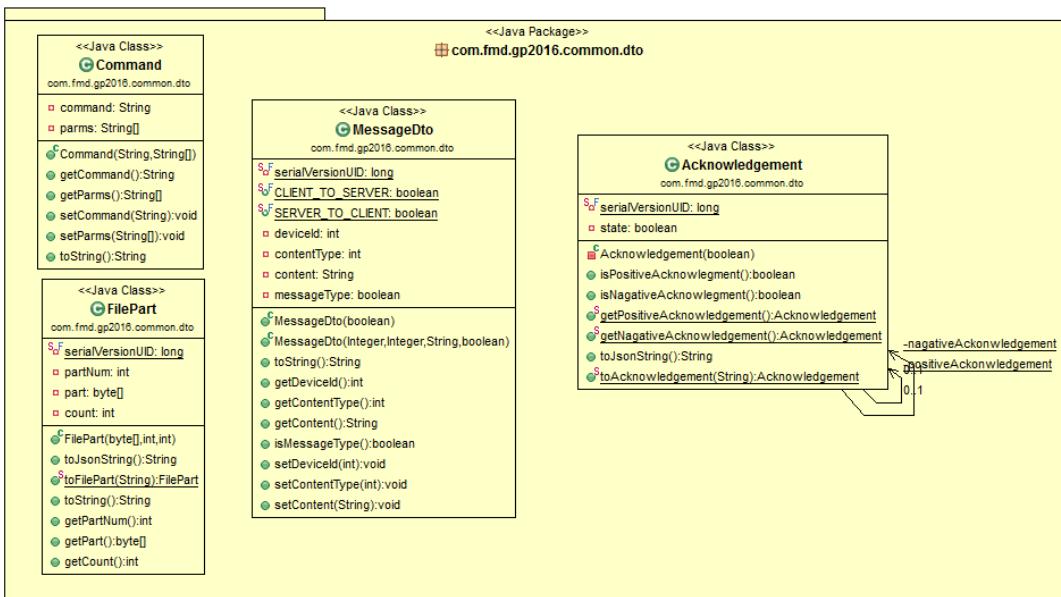
Whole view



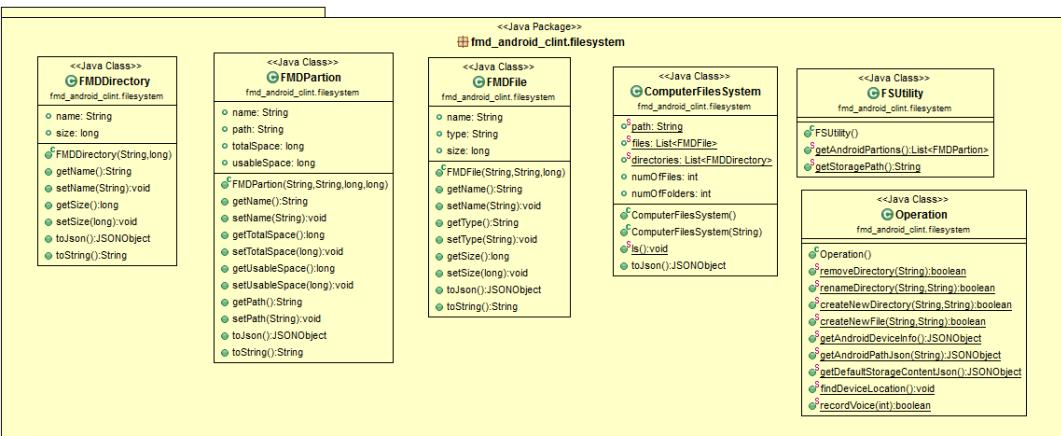
Helper methods



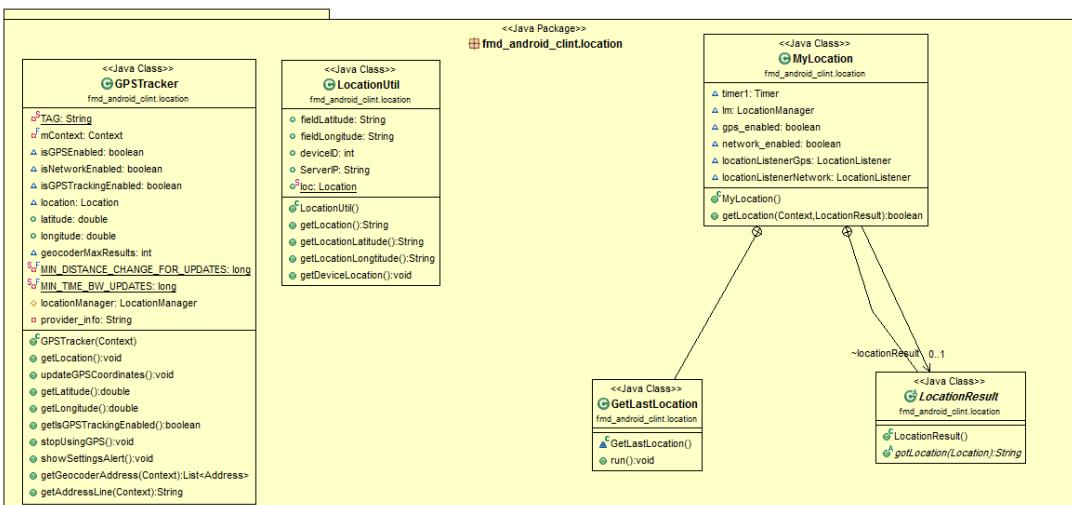
## Communication message



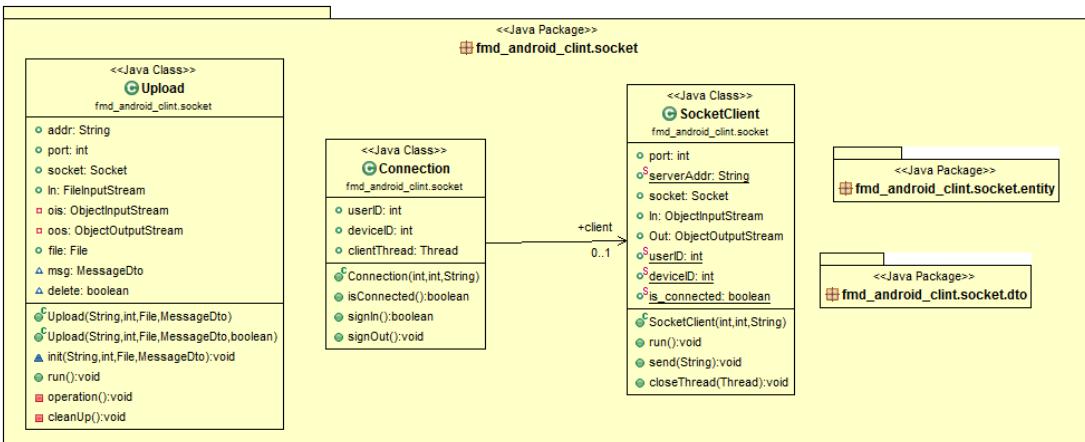
## File System



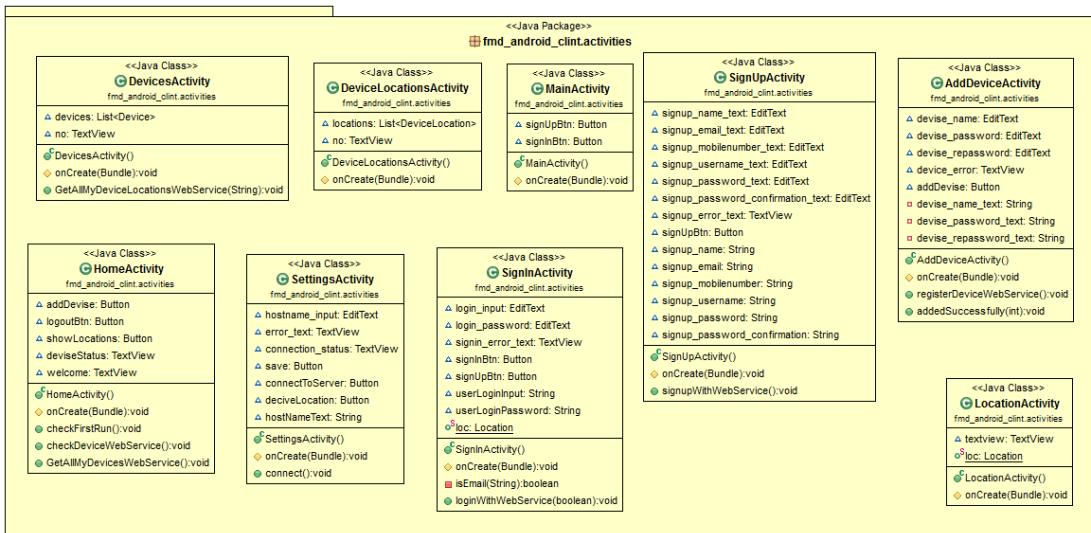
## Location



## Socket package

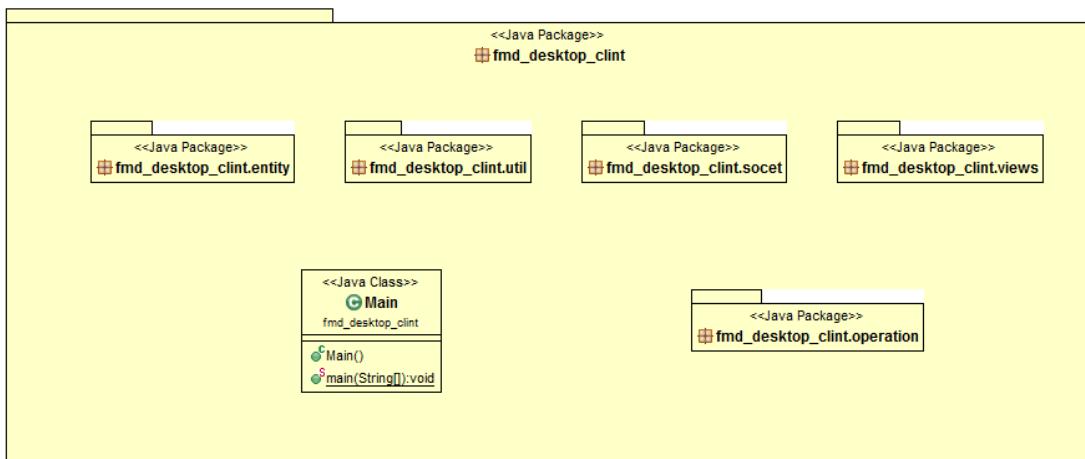


## View

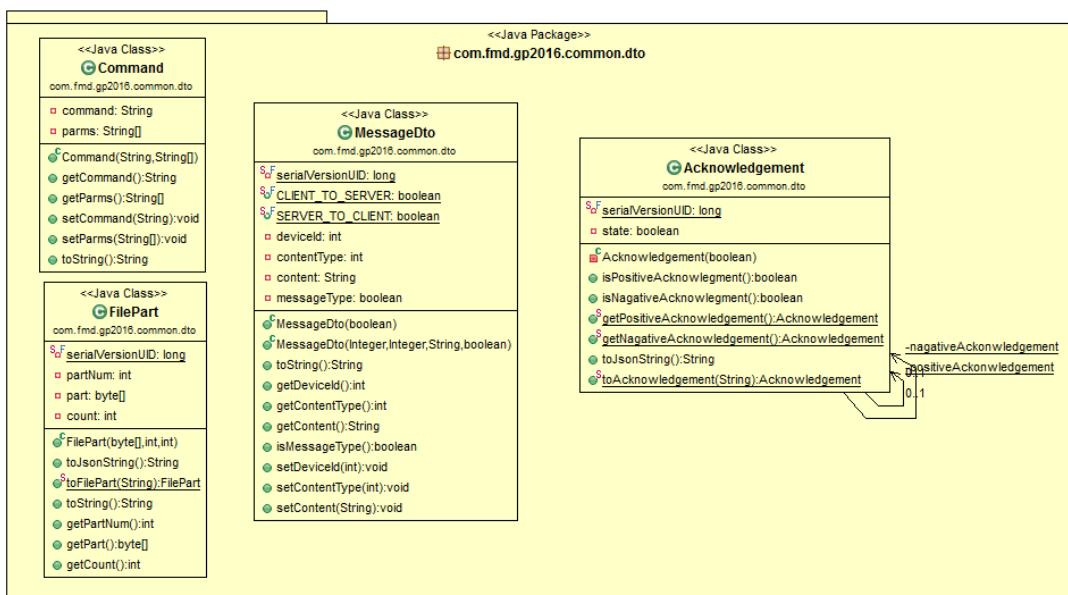


# Class Diagram for Desktop Application

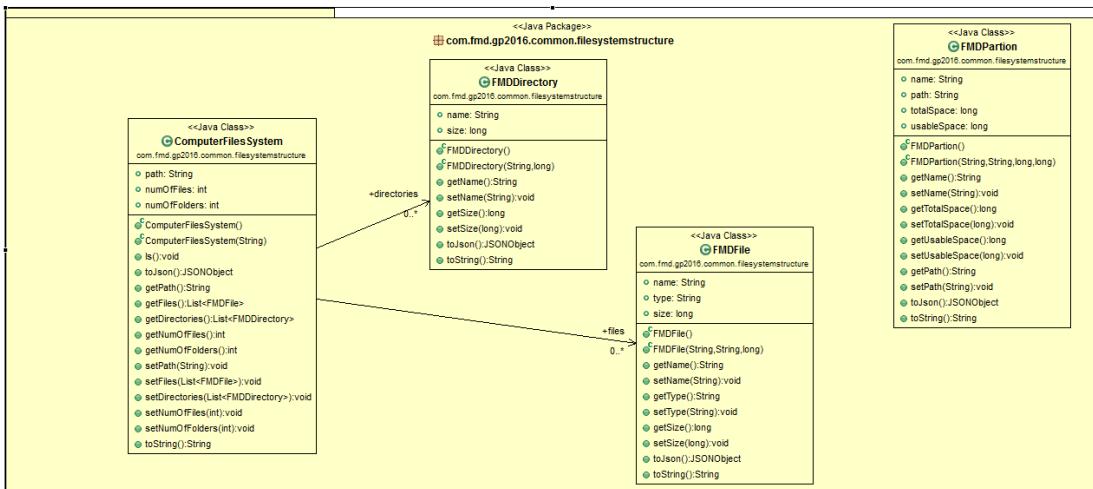
Whole view



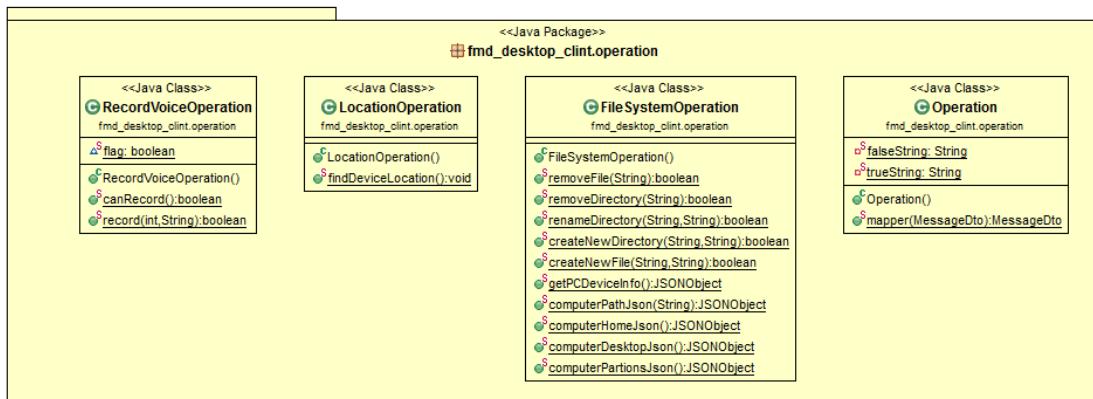
Communication message



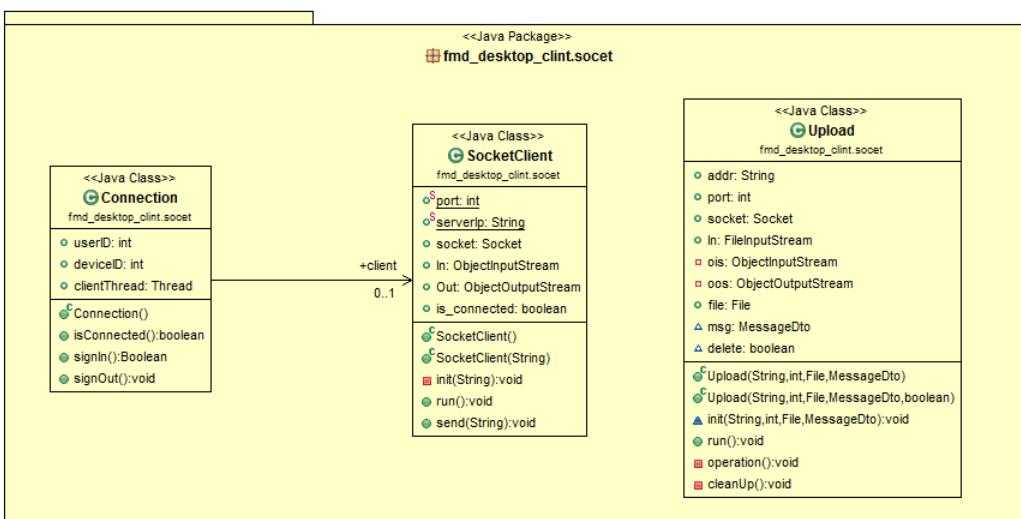
## File system structure



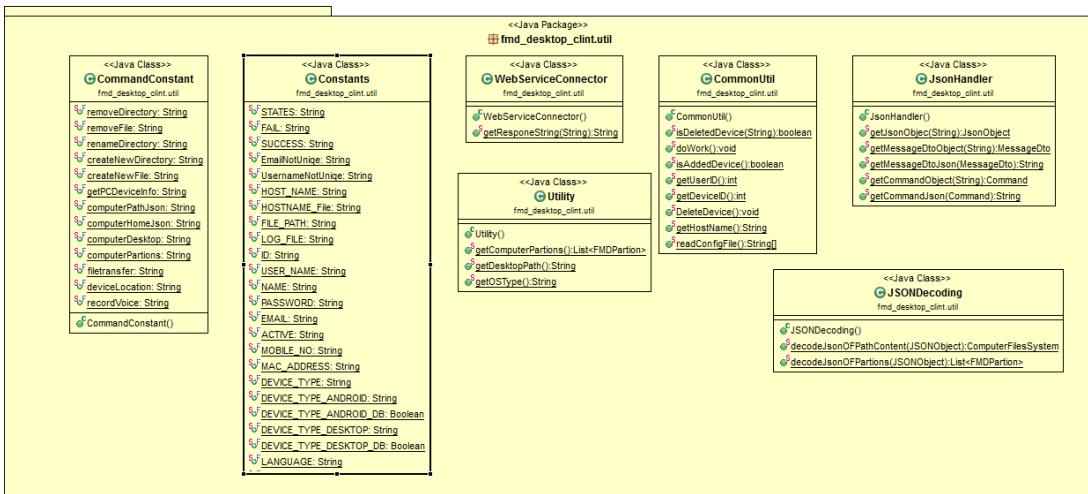
## Operations



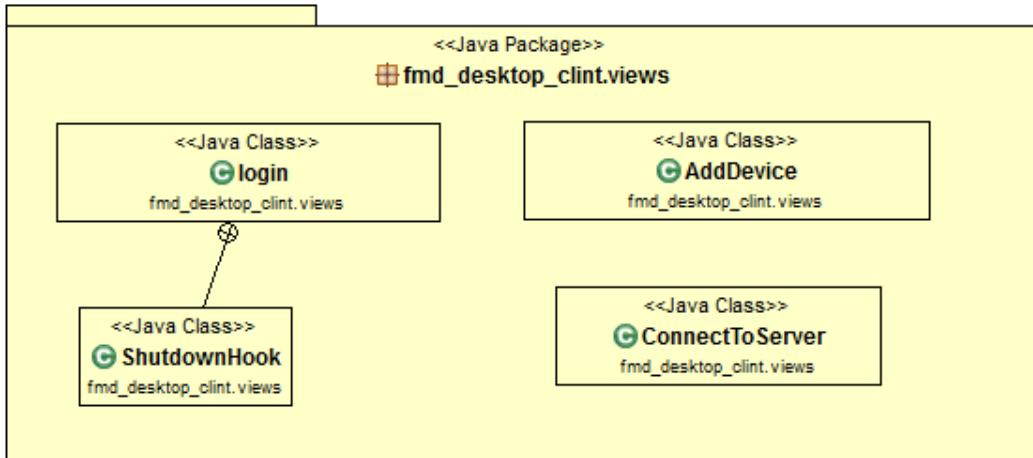
## Socket



## Utilities



## View



# Chapter 4

## Snapshots

## Web app Screenshot



We help people to access their lost or stolen electronic devices like tablet, mobile and laptops.

This access includes operations like (access file system storage, get file from the device, delete file, know the location of device, record voice, record video)

Username

Password

[New User ?](#) [Login](#)

[Download link](#)

FMD, Copyright © 2016 All rights reserved.



Name :

Mobile Number :

Username :

Email :

Password :

Confirm Password :

[Already have an account?](#)

[Sign up](#)

FMD, Copyright © 2016 All rights reserved.

Profile Page x ibrahim

localhost:8080/fmd/updateProfile.xhtml

Arabic English

# FIND MY DEVICE

Home Profile Downloads User Manual About Us logout



Name : Ibrahim Ali

Mobile Number : 01227462844

Username : test

Email : test

Edit Profile

FMD, Copyright © 2016 All rights reserved.

Profile Page x ibrahim

localhost:8080/fmd/updateProfile.xhtml

Arabic English

# FIND MY DEVICE

Home Profile Downloads User Manual About Us logout



Name : Ibrahim Ali

Mobile Number : 01227462844

Username : test

Email : test

Edit Profile

FMD, Copyright © 2016 All rights reserved.

Downloads Page x ibrahim

localhost:8080/fmd/downloads.xhtml

Arabic English

# FIND MY DEVICE

Home Profile Downloads User Manual About Us logout



Device Type: Android Mobile

Download link: [Find My Mobile](#)



Device Type: Desktop PC

Download link: [Find My Desktop](#)

User Manual Page

localhost:8080/fmd/usermanual.xhtml

Arabic | English

# FIND MY DEVICE

Home | Profile | Downloads | User Manual | About Us | logout

**HOW DO I CREATE A NEW ACCOUNT!**

write your Name :  Name :   
 write your mobile number :  Mobile Number :   
 write your Username :  Username :   
 write your Email :  Email :   
 write your password :  Password :   
 write confirm password :  Confirm Password :

[Already have an account?](#) [Sign up](#) [press here to sign up](#)

[press here if you already have an account](#)

**HOW DO I LOGIN TO MY ACCOUNT IF I HAVE AN ACCOUNT!**

FMD, Copyright © 2016 All rights reserved.

About Us

localhost:8080/fmd/about.xhtml

Arabic | English

# FIND MY DEVICE

Home | Profile | Downloads | User Manual | About Us | logout

**Background**



As we live in modern era of speed and technology, everyone seeks to a fast access to the information and having multiple devices containing different data. The user now become in a need of a new way to have all this information instantly wherever it is in which device it is. By that, his life routine will be much simpler. Furthermore availability of the Internet everywhere and at high speed and low cost will change the shape of life, so we think that building our graduation project will facilitate the communication with your devices even you are away.

**Problem Definition**



We help people to access their lost or stolen electronic devices like tablet, mobile and laptops, via web app and pre-installed app on these devices.

FMD, Copyright © 2016 All rights reserved.

Devices Page

localhost:8080/fmd/userDevices.xhtml

Arabic | English

# FIND MY DEVICE

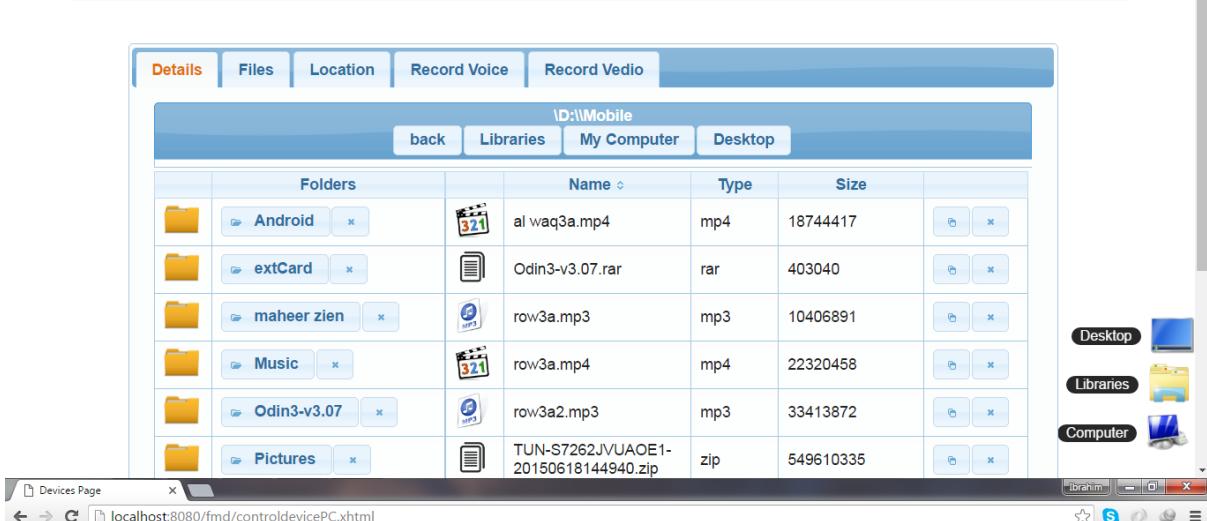
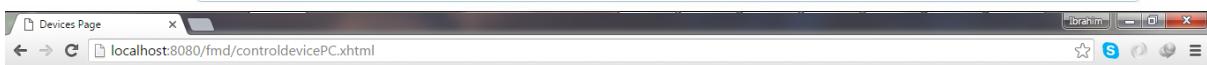
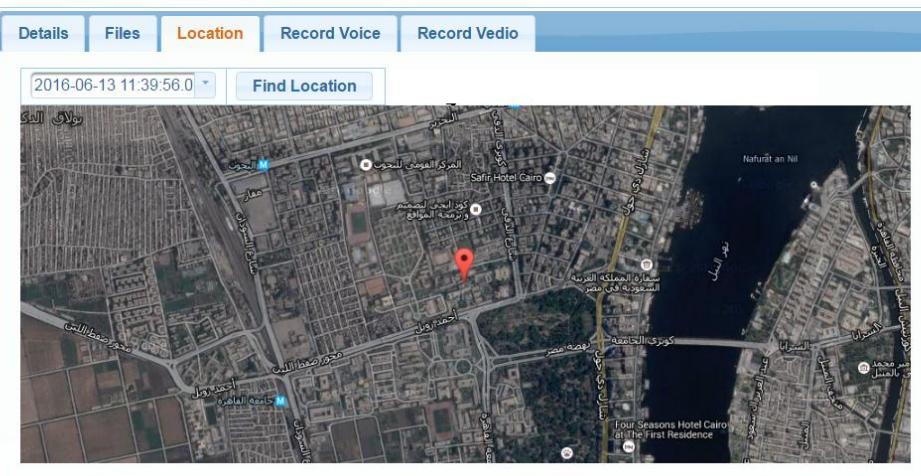
Home | Profile | Downloads | User Manual | About Us | logout

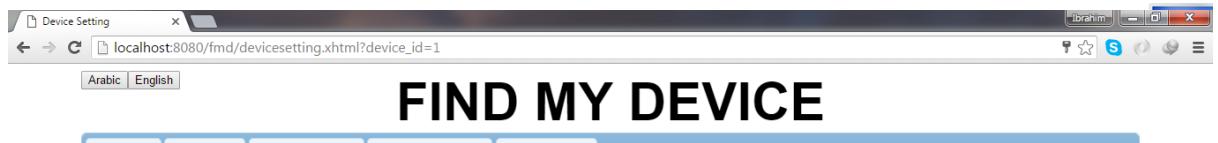
All your devices				
...	Device Name	Last Active Date	Device Type	Operations on Device
	asd		Now	 

FMD, Copyright © 2016 All rights reserved.

[Arabic](#) [English](#)

# FIND MY DEVICE

[Home](#)[Profile](#)[Downloads](#)[User Manual](#)[About Us](#)[logout](#)



Device Setting × librahim

localhost:8080/fmd/devicesetting.xhtml?device\_id=1

Arabic English

# FIND MY DEVICE

Home Profile Downloads User Manual About Us logout

Details	Change Password	Change Name	Files	Location	Configuration	Commands
Device Name	<input type="text" value="test"/>					
Password	<input type="password" value="****"/>					
<input type="button" value="Change"/>						

FMD, Copyright © 2016 All rights reserved.

Device Setting × librahim

localhost:8080/fmd/devicesetting.xhtml?device\_id=1

Arabic English

# FIND MY DEVICE

Home Profile Downloads User Manual About Us logout

Details	Change Password	Change Name	Files	Location	Configuration	Commands
Response Time	<input type="text" value="6"/>					
Video Record Time	<input type="text" value="20"/>					
Audio Record Time	<input type="text" value="20"/>					
<input type="button" value="Change"/>						

Device Setting × librahim

localhost:8080/fmd/devicesetting.xhtml?device\_id=1

Arabic English

# FIND MY DEVICE

Home Profile Downloads User Manual About Us logout

Details Change Password Change Name Files Location Configuration Commands

No pending operations now

delete

Devices Page × librahim

localhost:8080/fmd/controldevicePC.xhtml

Arabic English

# FIND MY DEVICE

Home Profile Downloads User Manual About Us logout

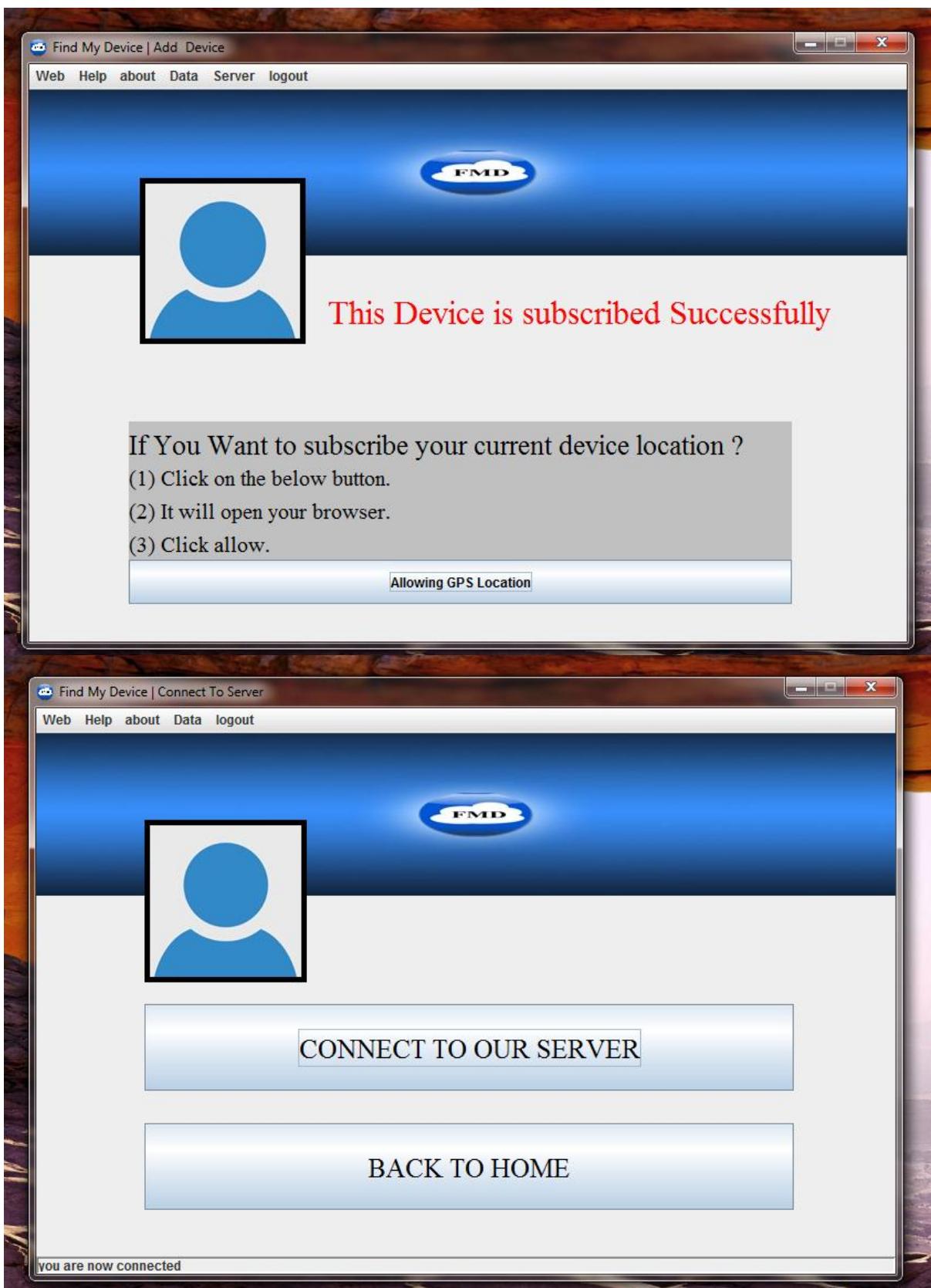
Details Files Location Record Voice Record Vedio

- Draft long document 2016\_06\_20\_15\_46\_57.docx

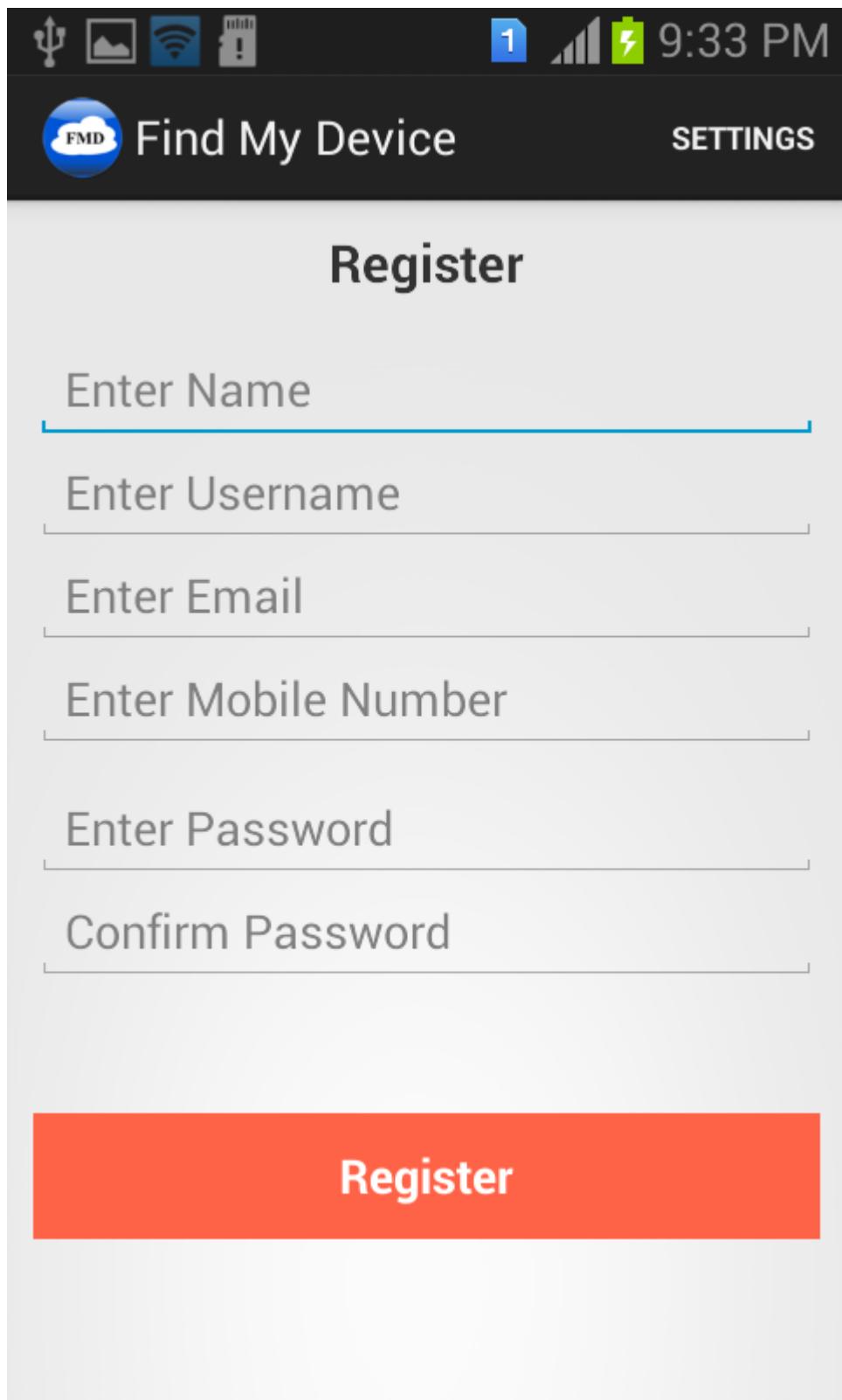
---

## Desktop App





## Android App





1



9:33 PM



Find My Device

SETTINGS

## Sign In

Enter Email or Username

Enter Password

**Sign In**

**Do not have an account ?**



1



9:34 PM



Find My Device

SETTINGS



Welcome  
hema



**View Your Devices  
Locations**

**Add This Device**

**Logout**



1



9:12 PM



Find My Device

SETTINGS



# Welcome hema

**This Device was registered  
successfully, you can access  
it remotely now...**



**View Your Devices  
Locations**

**Logout**



1



9:13 PM



Find My Device

SETTINGS

## Server IP

192.168.43.162

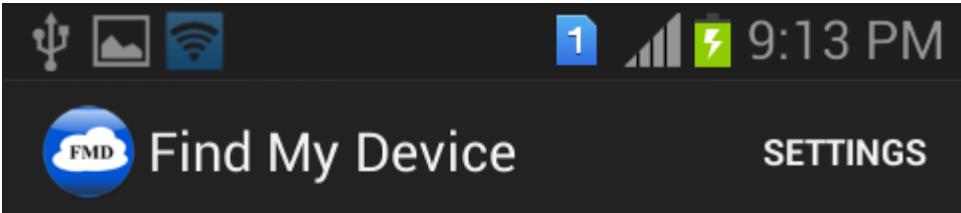
Save

## Server Connection

Connect To Server

## Connection Status

Status : Connected





1



9:09 PM



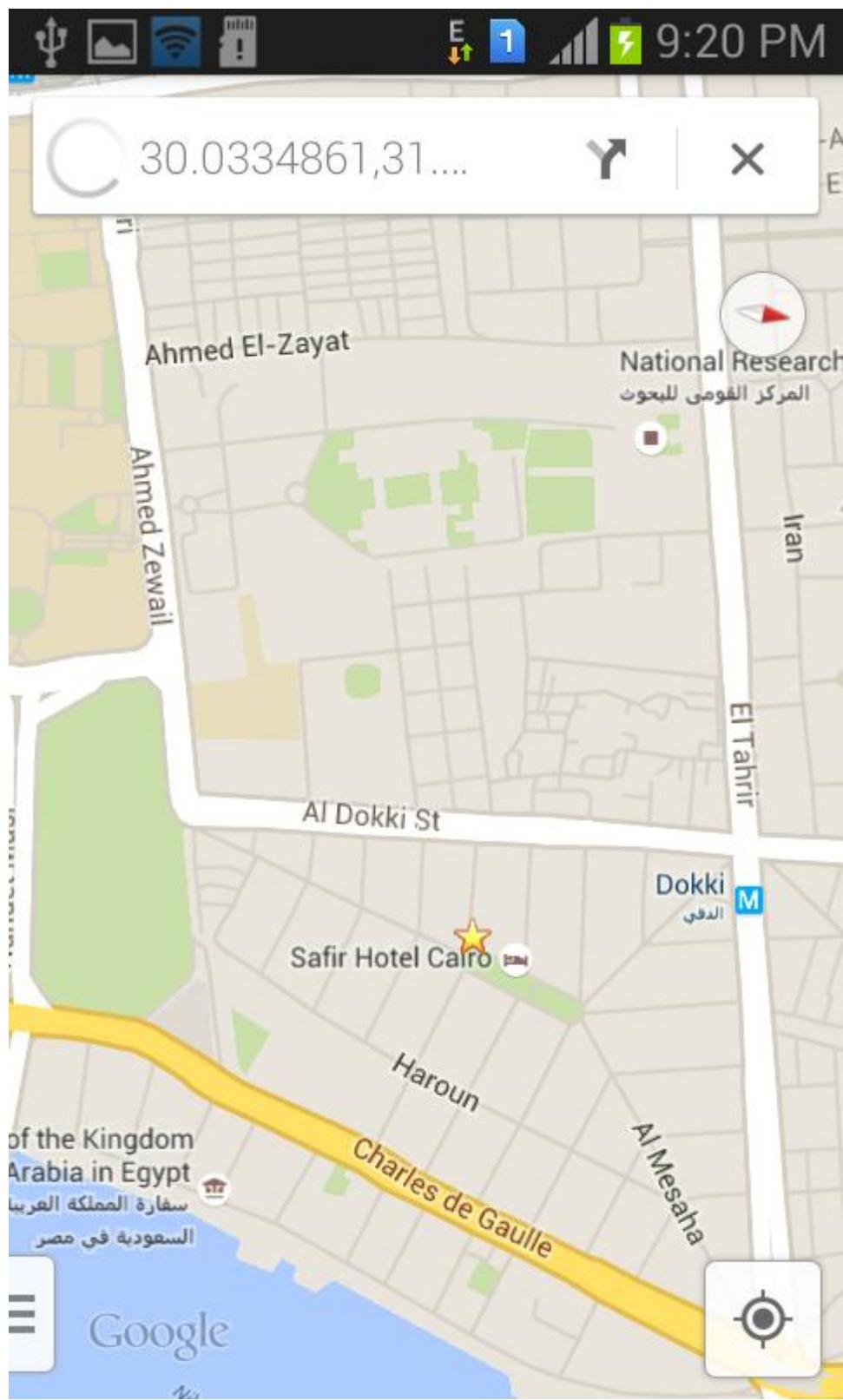
Find My Device

SETTINGS

## Device Loc History

04/27/2016 21:07:08

04/27/2016 21:09:21



# Chapter 5

## Summary

## Summary

---

Find my device has succeeded to meet its functional requirements, Server app and clients are fully developed and communicate properly.

The following is a summary about project functional requirements and states

1. Navigate file system
  - a. Based on socket programming
  - b. Sending commands to client and receiving JSON response
2. Find location
  - a. Based on network provider in desktop client and GPS or network provider in android client
  - b. Accuracy of location depends on accuracy of network provider
  - c. Saving device location manage to track device (history)
3. Recording (audio and video)
  - a. Based on availability (microphone and camera) in device
  - b. Audio and video in desktop client, audio in android client
  - c. Acceptable audio and video
  - d. Configurable audio and video recording time
4. Transfer files
  - a. Using fault handler approach to prevent file corruption during transferring
  - b. Saving files on server and can be downloaded anytime
5. Delete files
  - a. Sending delete command to device, then device perform command
6. Offline communication
  - a. Based on storing image of specific path (when open it) at point of time.  
When device offline, System preview stored image and manage you to store file transfer or delete command for specific file in image (will be sent to device when be online)

## References

- 1) Socket programming: [www.java-world.com/jaworld/jw-12-1996/jw-12-sockets.html](http://www.java-world.com/jaworld/jw-12-1996/jw-12-sockets.html)
- 2) Socket programming: <http://www.codeproject.com/Articles/524120/A-Java-Chat-Application>
- 3) Android development:  
[https://www.youtube.com/watch?v=QAbQgLGKd3Y&list=PL6gx4Cwl9DGBsvRxJJ0zG4r4k\\_zLKrnxl](https://www.youtube.com/watch?v=QAbQgLGKd3Y&list=PL6gx4Cwl9DGBsvRxJJ0zG4r4k_zLKrnxl)
- 4) Hibernate: <https://www.youtube.com/watch?v=wNT-EZsaC98>
- 5) Exe setup: <https://www.youtube.com/watch?v=wP9HcD4oPDw>
- 6) Spring: [www.tutorialspoint.com/spring/](http://www.tutorialspoint.com/spring/)
- 7) Spring: [www.javatpoint.com/spring-tutorial](http://www.javatpoint.com/spring-tutorial)
- 8) Spring: [www.javabeginnerstutorial.com/spring](http://www.javabeginnerstutorial.com/spring)
- 9) Java Swing: [www.tutorialspoint.com/swing/](http://www.tutorialspoint.com/swing/)
- 10) JSF: <https://www.youtube.com/watch?v=6jSzVlpyINQ>

## Code Repository

Desktop repo: [https://github.com/GraduationProject2016/Desktop\\_Version](https://github.com/GraduationProject2016/Desktop_Version)

Android repo: <https://github.com/GraduationProject2016/AndroidProject>

Web app repo: <https://github.com/GraduationProject2016/Find-My-Device>

All codes: <https://github.com/GraduationProject2016/>