**1.Registration via email:**

Successful Scenario:

1.Enter the email address.

2. Enter the username.

3. Enter password.

4. Enter the password again to confirm.

5. Enter other requirements like name, gender, age etc.

6. Agree to terms and condition.

7. Press submit button.

8. Confirm account by opening the link sent on gmail.

|  |  |  |
| --- | --- | --- |
| Class name | attributes | methods |
| Reg\_email | email Id,  Username,  Password,name, | store\_info()  send\_link()  confirm() |

**2. Registering via mobile number:**

Successful Scenario:

1.Enter the mobile number.

2. Enter the username.

3. Enter password.

4. Enter the password again to confirm.

5. Enter other requirements like name, gender, age etc.

6. Agree to terms and condition.

7. Press submit button.

8. Confirm account by entering the OTP sent to the entered mobile number.

|  |  |  |
| --- | --- | --- |
| Class name | attributes | methods |
| Reg\_mob | mobile\_no  Username,  Password,name, | store\_info()  send\_OTP()  confirm() |

**3. Login:**

Successful Scenario:

1. Entering of valid username and password.

|  |  |  |
| --- | --- | --- |
| Class name | attributes | methods |
| Login | Username,  password | authenticate() |

**4.Information Storing:**

Successful Scenario:

1.User registered on application.

2.User login successfully

3.User enter information asked.

|  |  |  |
| --- | --- | --- |
| class | attribute | Methods |
| User | Username  Blood group  Allergies  blood -pressure  Health problems | strore\_info() |

**5. Activation of application:**

Successful Scenario:

1.Activation of application by pressing emergency button.

|  |  |  |
| --- | --- | --- |
| class | attributes | methods |
| Emergency Mode | status | update(), delete() |
| confirm\_emergency | status | send\_OTP(),  check() |
| capture | capture\_status | capture\_location(),  capture\_photos(),  store() |

**6. Selecting the user**

Successful Scenario:

1. Open application directly.
2. Option is provided to choose the user for whom you want emergency service.
3. Select one option either other or myself.

|  |  |  |
| --- | --- | --- |
| class | attributes | methods |
| Dial\_manual | User\_type, service\_option, | read\_location(), search(), display(), send\_captured\_info() |

**7. Selecting the emergency service**

Successful Scenario:

1. Open the application.
2. Select the user from the option provided.
3. Enter the GPS location if other is selected.
4. Select the services you want like ambulance, fire brigade, or police etc.

|  |  |  |
| --- | --- | --- |
| class | attributes | methods |
| Help centre | Name,  acceptance\_status | send\_status()  receive\_request()  send\_help() |

**8.Calling Hospital:**

Successful Scenario:

1.User open application manually.

2.User select call hospital option from menu.

3.User get contact number and location of nearby hospitals.

4.User select hospital.

5.User call hospital or send message to hospital

**9.Calling Police:**

Successful Scenario:

1.User open application manually.

2.User select call police option from menu.

3.User get contact number and location of nearby Police station.

4.User call Police Station.

**10. Calling Fire Brigade:**

Successful Scenario:

1.User open application manually.

2.User select call fire brigade option from menu.

3.User get contact number and location of nearby Fire Brigade.

4.User call Fire Brigade station

Hospital, Police station and fire station are type of help centre and no extra functionality is needed. So we will keep only one class help centre and will keep a type attribute in this class which will store the type of help needed. So our final help centre class is:

|  |  |  |
| --- | --- | --- |
| class | attributes | methods |
| Help centre | Name,  Acceptance\_status,  type\_of\_service | send\_status()  receive\_request()  send\_help() |

By performing the Walkthrough we got the following extra classes:

**Missing classes:**

|  |  |  |  |
| --- | --- | --- | --- |
| Class | Attribute | Methods | Reason |
| Dial\_manual | User\_type, service\_option, | read\_location(), search(), display(), send\_captured\_info() | This class is useful when the user wants the emergency service for someone else or when he wants service other than the ambulance. |
| Confirm\_emergency | status | send\_OTP(), check() | This class is to confirm emergency mode.if user has pressed panic button by mistake then user can shut down emergency mode, so further process will not take place. |
| Registration | Name, username, password | store\_info(), confirm(), send | This class is needed to make account of the user on the system. |
| Login | Username,  password | authenticate() |  |

Reg\_mob and Reg\_email classes are inherited from the Registration class. Reg\_email will inherit all the attributes and methods of the Registration class and will have email\_id attribute and send\_link() method of its own. The Reg\_mob class will also inherit all the attributes and methods of Registration class and will have mob\_no attribute and send\_OTP method of its own.

So the final classes are:

|  |  |  |
| --- | --- | --- |
| Class | attribute | methods |
| Emergency Mode | status | turn\_on(),  turn\_off()  get\_status() |
| User | Username,  Blood\_group,  Blood\_pressure,  Health\_problems | store() |
| Auto\_dial | contact\_number,  help\_centre\_contact | auto\_search(), call(), send\_SMS() |
| Help centre | Name,  Acceptance\_status,  Type\_of\_service | send\_status()  receive\_request()  send\_help() |
| capture | capture\_status | capture\_location(),  capture\_photos(), store() |
| Emergency\_contact\_list | contact\_number,  priority | update(),delete() |
| Dial\_manual | user\_type, service\_option, | read\_location(), search(), display(), send\_captured\_info() |
| Confirm\_emergency | status | send\_OTP(), check() |
| Registration | name,  username, password,  email Id,  Mobile no | store\_info(), confirm(), send\_OTP(), send\_link() |
| Login | username,  password | authenticate() |

