



# **Project Assignment for the Course on Programming for Ubiquitous Things Spring 2019 IFI/UiO**

## **AutoSure - an app for auto insurance Testing/Demonstration Roadmap**

This document is intended to help students when testing their project. Note that these are non-exhaustive tests. Further actions may be requested by the teacher depending on the implementation decisions of certain features. Students are advised to test their project thoroughly.

### **Scenario Setup:**

- 2 users present in the server state with at least one claim and one claim message for each.

### **Basic Features**

- **Test 1: User Login, Logout and detailed information**

Setup: Load server with sample users

- Step 1: Login with a user (**user1**) using the **correct** id and **incorrect** password  
Expected output: login should **not** succeed
- Step 2: Login **user1** with **correct** id and **correct** password.  
Expected output: login should succeed
- Step 3: Display **user1** detailed information
- Step 4: Logout and show that the user is no longer allowed to interact with his account  
Expected output: logout should succeed
- Step 5: Login with a different user (**user2**) using the **correct** id and **correct** password.  
Expected output: login should succeed.
- Step 6: Display **user2** detailed information

- **Test 2: List Claims.**

- Step 1: List the claims for the logged in user  
Expected output: a list-form of claims available.
- Step2: Select a claim  
Expected output: the claim detailed information should be presented to the user

- **Test 3: Submit new Claim**
  - Step 1: Select the option to file a new claim
  - Step 2: Fill in all the claim details, selecting one of the available number plates
  - Step 3: Submit the new claim
  - Step 4: List claims  
Expected output: the new claim should be present in the list
  - Step 5: Select the fresh issued claim  
Expected output: the claim detailed information should be presented to the user
- **Test 4: Claim Messages**
  - Step 1: Show how a user can view the messages of a particular claim
  - Step 2: Create a new message
  - Step 3: Fill in the message details
  - Step 4: Send the message
  - Step 5: List the messages for that claim  
Expected output: the new message sent should be shown
  - Step 6: Allow to see the full details of a message for that particular claim
  - Step 7: Show how a user can view the messages of a **different claim**  
Expected output: a different list of messages should appear

## **Advanced Features**

- **Test 5: New Message Notification**
  - Step1: In the server, issue a new message to the user logged in  
Expected result: a form of notification should appear in the emulator informing the arrival of the new message.
- **Test 6: Offline Mode**
  - Step 1: In the emulator with the application running, disconnect the network
  - Step 2: Show that partial information accessed before step 1 is still available offline.
  - Step 3: Fully close the application
  - Step 4: Open the application again and show that partial information accessed before step 1 is still available offline.
  - Step 5: In the emulator, connect the network

- Step 6: Logout
- Step 7: Login with a user different from the previous user
- Step 8: In the emulator, disconnect the network
- Step 9: View the customer information  
Expected output: the information of the previous user logged in should **not** be displayed