Use Cases and Logical Architecture

* XID: X00126628
* Name: Tudor Voda
* Project Title: Weather Bot

**Provide at least 6 Use-cases describing the functionality of the proposed system**

## Section 1: For Each Use Case:

|  |  |
| --- | --- |
| Use Case: | Display weather |
| Actor(s): | User |
| Goal: | The customer gets to see the details about the weather |
| Overview: | The customer accesses the applications, they type in their location and the bot connects to the API and the bot displays the data in a format of an Adaptive Card. |
| Pre- Condition: | Locations exists, Correct name entered. |
| Post- Condition: | Weather display in an Adaptive Card. |
| Successful Scenario: | 1.User opens the application.  2.Enters the locations they want to see weather information.  3.The applications connects to the API and displays the relative information to the user. |
| Alternative Scenario(s): | 1. User opens the application.  2. Enters the locations they want to see weather information.  3. The applications can’t find the location it may be invalid. |

|  |  |
| --- | --- |
| Use Case: | User location |
| Actor(s): | User |
| Goal: | The user clicks a button to input their location without them typing it |
| Overview: | The user clicks the location the button, the application will enter their location in the input box. |
| Pre- Condition: | The user allows location on the browser. |
| Post- Condition: | The application has the location of the user. |
| Successful Scenario: | 1.User allows location on the browser.  2.User clicks the Location button.  3.The application get the user location.  4. Application displays user location. |
| Alternative Scenario(s): | 1. The user has location block on the browser.  2.Application cannot get location. |

|  |  |
| --- | --- |
| Use Case: | Register |
| Actor(s): | User |
| Goal: | The user created an account for the application |
| Overview: | The user will click on the registration button on the navigation bar. The user will enter their email, password and confirm password. |
| Pre- Condition: |  |
| Post- Condition: | User has created an account. |
| Successful Scenario: | 1. The user goes onto the registration page.  2. Enters a valid email, password and confirm password.  3. The application registers the user to the database. |
| Alternative Scenario(s): | 1. The user goes onto the registration page.  2. User enters an email already in the database.  3. User unsuccessful registration to the database. |

|  |  |
| --- | --- |
| Use Case: | Login |
| Actor(s): | User |
| Goal: | The user login on the application. |
| Overview: | The user will click on the login button on the navigation bar. The user will enter their email and password. |
| Pre- Condition: | User has an account in the database. |
| Post- Condition: | User has access to their account. |
| Successful Scenario: | 1. The user goes onto the login page.  2. Enters their email and password.  3. The application checks the credentials against the database  4. User is given access to their account. |
| Alternative Scenario(s): | 1. The user goes onto the login page.  2. Enters their email and password.  3. The application checks the credentials against the database.  4. User credentials are incorrect.  5. User is denied access. |

|  |  |
| --- | --- |
| Use Case: | Access the Database |
| Actor(s): | Admin |
| Goal: | Admin access the database to add, remove or change the entries in the database |
| Overview: | The admin login the database which will allow the admin to add, remove or change any data that is stored in the database |
| Pre- Condition: | Have an admin account |
| Post- Condition: |  |
| Successful Scenario: | 1.Admin login to the applications  2.The admin connects to the database  3.The admin can add, remove or change the database |
| Alternative Scenario(s): | 1.Admin login to the applications  2.The admin connects to the database  3.The login details enter are wrong, refuse connection |

|  |  |
| --- | --- |
| Use Case: | User updates their profile |
| Actor(s): | User |
| Goal: | User updates their favorite quote |
| Overview: | The user accesses their account and updates their quote. |
| Pre- Condition: | User has an account. |
| Post- Condition: | User has a new quote. |
| Successful Scenario: | 1. The user logins in the application. 2. Goes to their dashboard. 3. Types the new quote and updates the quote |
| Alternative Scenario(s): | 1.User enters invalid credentials.  2.Application denies access to the user. |

## Section 2: Logical Architecture

Software Components: UI connects to the Microsoft bot, with connection to the database

Databases: MongoDB

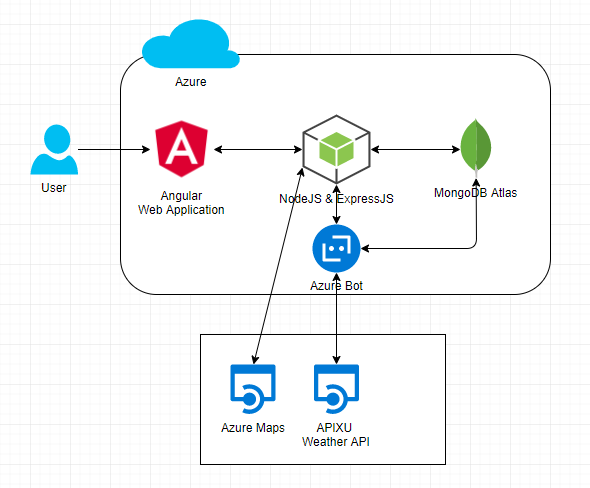
App engines: Angular 7, Microsoft bot framework

Mobile platforms: All

API’s and languages: APIXU Weather API, Azure Maps, Typescript, NodeJS and ExpressJS

Deployment EG RESTful JDBC session less: RESTful

Security e.g. Https certifications authentication: App services provides authentication



## Logical Architecture Discussion

The user accesses the application that is deployed in azure. The use can use the webchat to ask the bot what the weather is. The bot will connect to an API from the backend to retrieve the data. Data is in json format and the application will display the request information to the user in the frontend. The weather is displayed on a Microsoft Adaptive Card to the user. The user can click get location button for the application to type in their location. The application is connect to Azure maps to get the user location using longitude and latitude. The user can also register an account to the database, allowing them to edit their profile.