Alabama Liquid Snake

University of Pretoria

Epi-Use

Botic - Privacy aware chatbot
Implementation Details

Justin Grenfell - u16028440
Peter Msimanga - u13042352
Alicia Mulder - u14283124
Kyle Gaunt - u15330967
Lesego Mabe - u15055214

Contents

1 Introduction

This documents follows the implementation decisions made during the project development stage. Implementation decisions will be covered in each interation, and categorized further according to the use cases.

1.1 Interation 1

1.1.1 UC21: Login

The Angular SPA application framework does not necessarily have a direct mapping to our architectural designs and decisions; as to be expected. The View will be bundled into varous Angular components. The Controllers will be bundled into various Angular services. The Model will also be represented as Angular services. This is how we will map each of the Angular constructs to the architecture we have chosen.

The Design Class Diagram represents certain constructs that are not necessarily Angular or Typescript contructs. Since we have made the decision to represent all our controllers as Angular services, we will have to create an Interface to represent the Abstract Controller class. Each Concrete Controller will thus be represented as an "implemented" Angular Service; [1] provides a tutorial on how to do this.

The "shared" folder will contain classes and objects that are shared across the system, i.e. log classes, data structures, and others.

2 References

[1] G. Marlow, “Creating interfaces for angular services,” https://medium.com/hackernoon/creating-interfaces-for-angular-services-1bb41fbbe47c, October 2017, accessed on 2019-08-04.