Human brain

team one

# Overview

Simulation of a human brain is a topic of interest in our days. Our team is going to create a service of the brain that is going to manage the services that scan for images and text. After that with the information achieved over time is going to take a decision.

## The Objective

* Creating a service that will manage the brain subservices.
* Creating a memory for our brain.
* Creating an interface for our services.

## Team Structure

Name - GitHub account - Role

* Munteanu Andrei-Stefan - MunteanuAndreiStefan - Scrum Master
* Costandache Mihai-Andrei - andreicostandache - Developer
* Zaharia Raul - rzaharia - Developer
* Ouatu Bogdan-Ioan - ilikehaskell - Developer
* Ghiga Claudiu-Alexandru - claudiu-ghiga - Developer
* Silistru Alexandru - SilistruAlexandru - Developer and Tester
* Ninicu Cristian - DoubleNy - Developer
* Marcu Alexandru - - Tester
* Dodu Emanuel-Andrei - - Tester

## Stakeholders

* University Alexandru Ioan Cuza, Faculty of computer science

# Requirements

## High Level Requirements

* The project will have only **one type of actor** identified by stakeholders:

* The **user** should be able to **see the memory** of the application and **upload a new image / text** and get a response for the upload. The response will be **concept** page or it will ask the **user to provide information** about the concept. The user will also be able to select a concept from the memory and modify / delete the concept.

## Detailed Requirements

* The project will have only **one type of actor** and he will can interact with our application:

* The **user** should be able to **see the memory** of the application.
* The **user** should be able to **select from the memory** a concept.
* The **user** should be able to **view / modify / add / remove** a concept. A concept will have a short text description and an image also a concept will have all the data sent we marked to be from current concept.

From view the **user** should be able to **generate content** that is like the one that is concept.

* The **user** should be able to upload a file text or image and get a response. A **response** is based on the **decision** our module service.
* The **decision** may be **to integrate the uploaded file into a concept** or **ask the user to create / edit a concept,** also an **not safe decision should be shown** here.
* The **user** should be able to **export** his **memory** to a file or **import** a memory.
* Each **memory** we use will be **secured** with a **password,** the user should provide a login password before interacting with a loaded memory.

## High Level Project Approach

* Methodology:
* We will have LSD based methodology adapted to our needs. We will make a merge between FDD and LSD.
* Overview of our methodology:
  + Eliminate waste
  + Amplify learning.
  + Develop overall model.
  + Build feature list.
  + Plan by feature.
  + Design by feature.
  + Build by feature.
  + See the hole.
* Information about our sprints and organization:
  + Sprints will count 1 to 10 days.
  + We will have a daily meeting or call every 3 days.
  + Sprint review / retrospective will take place as normal.
  + The backlog will be kept on GitHub.com
  + Tests will be written but will started by the programmers for each build. After the project is published the tests will be moved to Jenkins to automate the build process.
* High level architecture of the application:
* **Main application** will be write in **Java using Swing**. It will have **4 screens**:
  + In first screen the user will be asked to **load a memory file**, on load he will be asked to **input a password.**
  + In second screen the user will have a **view over all concepts** (memory) and a **button for upload**.
  + In third screen it will be the **concept details**, details may be modified inside this screen. In this screen the user
  + In fourth screen it will be the **result from the upload action**.
* **All the data received by our service is going to be written into JSON files.**
* The **service** that is going to take **decisions** is going to be written in **Java**, but it may call **multiples** **other services.** It may be implemented as a decision tree, but is not yet nailed. We may use Weka to simplify our work.
* The **service** that is going to **generate content** for a concept is going to be written in C++ and called into our application.
* In the **background** **after we call the image service** we may get the text description from it **and call the text service.**
* Both services for **Image and Text** will receive their input and will return a **JSON** file in which they will specify the **concept** found, the **matching** value in a **number from 0 to 1** and a **short description** if is possible. For a more detailed version please look into the json file provided.
* The **image module** will **create** a function that will receive a **buffered image** and will return to us the json.
* The **text module** will **create** a function that will accept a **string** and will return to us the json.
* **Both operations described above should be async.**
* To see an example of JSON please look into: https://github.com/MunteanuAndreiStefan/ProjectIP/blob/master/Architecture/Examples/JsonFormat.json

## Project Deliverables

Following is a complete list of all project deliverables:

|  |  |
| --- | --- |
| Deliverable | Description |
| The application solution | The project solution will be on a repository under an MIT License |
| The documentation of the project | Delivered to the end of the project to the teachers. |
| Research documents | All the documents we written by us to exchange information |