Student: Rupert Gobber
Student ID: 207554

Introduction to Service Design and Engineering

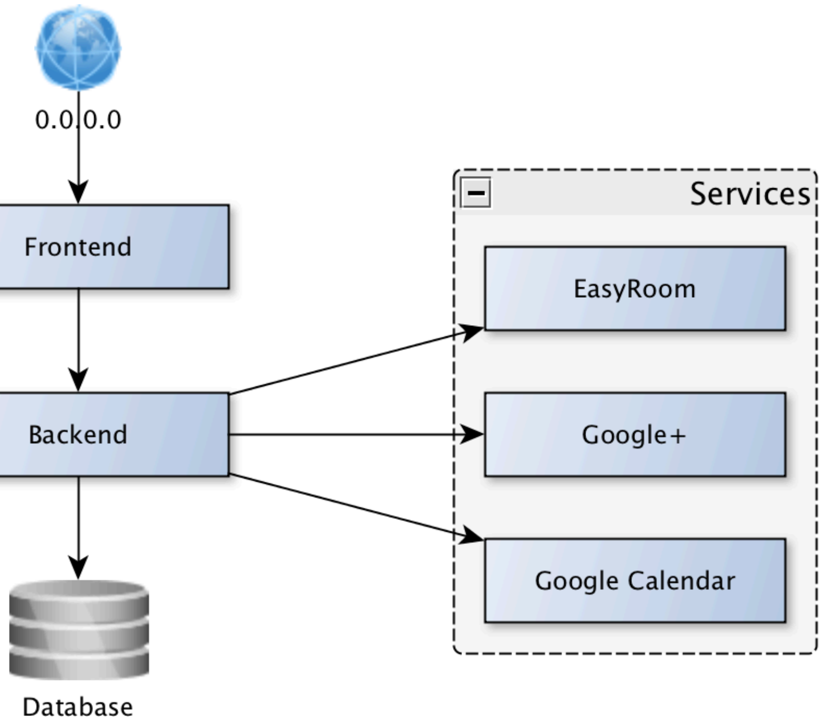
Project Report

Basic architecture

- Node.js application for backend, using Express to create a web service applications;
- Pug framework for frontend, with simple pages to simplify the users experience;
- MongoDB database
- 3 other external services

Components

- Google+ APIs, used to log people that use this webservice;
- Easyroom APIs, used to get in realtime the occupations of Povo rooms;
- Google Calendar APIs, used to ask permission to get events in specifics calendars;
- Mongodb APIs, used to store some data and semplifiyng the workflow application;



Dependencies

- Express.js, to create a routed web application
- Mongoose, to connect to mongodb
- Passport.js, to manage the tokens for google APIs
- Morgan, to log the Rest call
- Cookie session, to create a session
- Http error, to create 404 pages with informations in case of bad request

Project folder structure

- bin, with launcher configuration
- config, with passport configuration and keys
- models, with mongodb model tables
- public,
  - javascript, script to manage the services
  - stylesheets, with css stylesheets
- routes, with express routers file
- views, with pug frontend pages

User Interface Details:

There are many pages, described as follow:

- "Homepage", with some informations about the usage of this web applications
- "Login" page, accessible pressing the "Login" button. After Google login, you will rendered in the "Profile" page.
- "Profile" page, where you can see some informations about your account
- "Logout" page, visible after an user is logged, that allows you to close the session
- "Select Calendar" page, where you can insert the name of your calendar. After a calendar is selected, a button "See events" will appear. Press it to see the events retrieved by calendar APIs
  - Note: the blue calendar is called"primary" by default
- "Freerooms" page that call the Easyroom Api and retrieve the rooms occupation of Povo
- "Run Demo" page, that will perform the merge task of events and free room in order to create a table with events and room for each time slot. A time slot duration is half hour.

Deployment

- This application is deployed on Heroku at this link

Local usage

- To run this application in local, follow this instructions:
  1. Clone the repo
     ```
     git pull https://github.com/gobberr/ISDEproject.git
     ```
  2. Move to correct folder
     ```
     cd ISDEproject
     ```
  3. Install Node.js dependencies
     ```
     npm install
     ```
  4. Run the service
     ```
     npm start
     ```

Known restriction

- It seems that the university's wifi blocks the connections to mongodb. Use a hotspot to run this application in the university