COMP3207 - CLOUD APPLICATION DEVELOPMENT

SORT MY LIFE OUT - REPORT BRIEF (TEAM P)

RICHARD COOK, ELEANOR HAMILTON, AMMAAR TAWIL, THOMAS LAM, KYM HULBERT, CLAYTON JONES, JON HANCOX

Application Link - http://sort-my-life-out.appspot.com/

1 PROTOTYPE DESCRIPTION

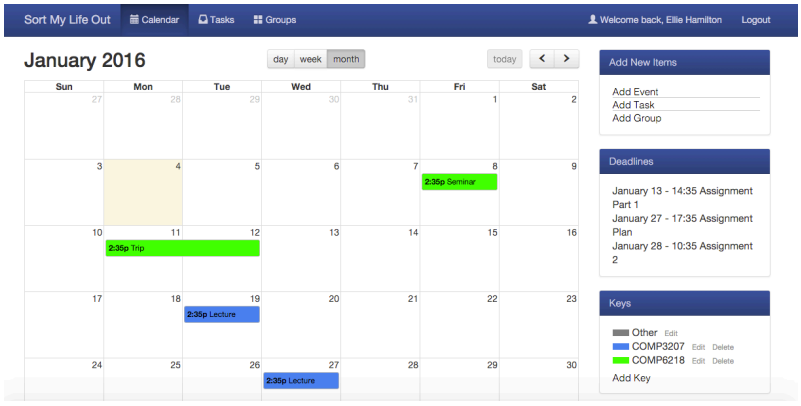
The aim of the application we designed was to improve the organisation of students at university. We chose this as the purpose of our design because many students struggle to be organised. We focused our application on students, as organisation is so important to this group as they often have to manage many responsibilities such as assignments, jobs and societies. Having a particular target audience for our application also allowed us to focus our design towards this audience (which we were very familiar with) in order to make our application as useful as possible.

We created an application that allowed users to manage a variety of aspects involved in student life. We implemented a calendar which students could use to add events including modules (lecture/seminars), shifts for a job, society events, one-off events etc. In addition to this, we included the functionality to manage tasks and deadlines (tasks with a date/time). These tasks and events could then be linked together using a key system, for example COMP3207 events and tasks could be set to green, making them easy to see at a glance.

Furthermore, we decided that another important feature to our audience was the need to manage group projects. In order to meet this need we implemented a group management function where users could create groups and be invited to groups by other users. Each group has a calendar of its own for managing group events such as meetings and also a task list in order to manage tasks and deadlines and assign these to each member.

Following the Waterfall model (explained in section 2), wireframes were designed and a specification was created. The final features chosen were:

- Create users
- Calendar – to manage timetable, modules etc
    - Adding and editing
    - Colour coding (using keys)
    - Different displays for the calendar – day, week, month
- Tasks
    - Ability to create deadlines – adding date/time
    - Tick off tasks
    - Colour coding (using keys)
    - First 3 deadlines in sidebar
- Groups
    - Add members
    - Add events to the calendar
    - Add tasks – assign to users
    - Edit group information
    - Ability to leave the group



2 TOOLS AND TECHNIQUES

TOOLS

- **Google App Engine**: The application was launched using Google App Engine (GAE), a platform as a service (PaaS) cloud platform. GAE has advantages such as automatic scalability of resources used depending on user demand, that results in good performance and reliability. The GAE NoSQL datastore is hosted on the GAE cloud platform and was used as the data storage for the application.
- **GitHub**: To allow all team members to work on the code for the application simultaneously, GitHub was used to help prevent conflicts in the code. Different parts of the application were developed at the same time, but often on the same document. Team members pushed and pulled their commits regularly to ensure that everyone had the most up to date versions of the different parts of the application to easily fix any conflicts. The code could be tested locally but some features required a connection to the server, requiring deployment to the live version for testing purposes.
- **Google Drive**: To develop ideas, gather requirements and manage any documentation, Google Drive was used so that all team members could simultaneously see and contribute to any work. It allowed all team members to access the documents from anywhere.
- **Cacoo**: Cacoo was used to design the front end of the application with wireframes and flow diagrams. All team members were able to access and contribute to the documents developed on this application.
- **Trello**: Used to list and assign tasks to team members, Trello proved an invaluable organisational tool to the team. It allowed information and ideas to be shared and tagged onto specific tasks that were visible to other team members working on them.
- **Facebook**: Facebook was used as the main tool for communication through the use of Groups and Messenger. Meetings were organised and planning for the project was discussed when the team was apart. This proved most useful as the team finished the project over the Christmas vacation.

TECHNIQUES

Waterfall Model: To develop the application, the waterfall model was employed. It involved sequentially completing tasks in stages by first developing the initial idea, requirements gathering, designing, implementing and then testing. It was decided that this would be more effective than developing the design whilst implementing it.

- **Bootstrap (http://getbootstrap.com/)**: By including Bootstrap, CSS coding was drastically reduced as Bootstrap includes styling and responsive sizing.
- **Spectrum Colour Picker (https://bgrins.github.io/spectrum/)**: There are several colour pickers available but the group had chosen a colour picker suitable to the design of the application.
- **Timepicker**, **Datepicker (bootstrap-datepicker.readthedocs.org/en/latest/, jdewit.github.io/bootstrap-timepicker/)**: To assist usability, timepicker and datepicker were utilised, which provided quality selection boxes for times and dates in JavaScript. This required testing the output after submission because it had to work with the backend of the application, which was already established and functional.
- **FullCalendar (http://fullcalendar.io)**: This addition to the application is one of the most involved as the whole application depends on the calendar implemented. By having this plugin, several features are already implemented, but customisation is required from plugins such as qTip (display information on hover) and Colourpicker (for the use of Keys).
  It was essential for the understanding of how some features were utilized in the application due to the additional features required in the specification.
- **qTip (qtip2.com/)**: FullCalendar does not offer a popup on hover or click of an event to display all the details of the event. As several features are available from this plugin, it was decided to choose a simple hover that would display information regarding the event or task created by the user.
- **Moment.js (http://momentjs.com/)**: This library is a powerful date manipulator and was used to display and manipulate dates in JavaScript. Implementing this functionality without the use of this plugin would result in more complex code and likely bugs and vulnerabilities.
- **HTML5BoilerPlate (https://html5boilerplate.com/)**: Using this template allowed for a fast and robust start with libaries such as jQuery, Bootstrap and Modernizer being included and referenced.

Language	Line of Code (LOC)
JavaScript	891
Python	1019
Misc	1353
Total	3263

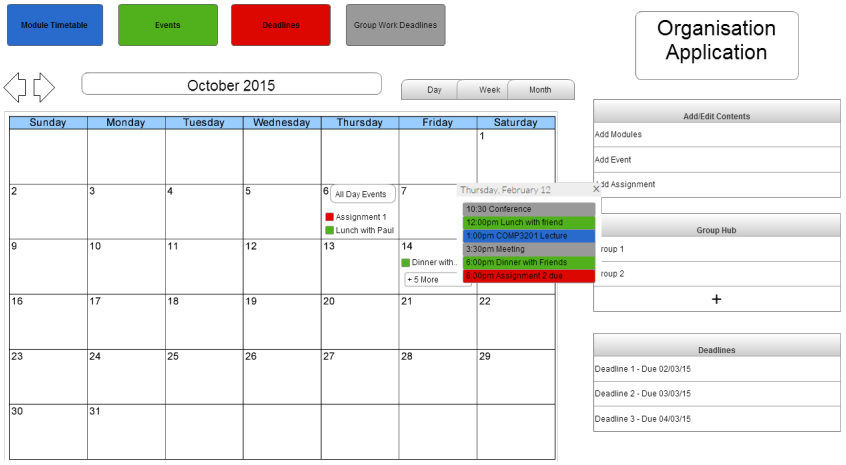
4 DESIGN AND IMPLEMENTATION

By executing the Waterfall model as discussed in section 2, this enabled the group to be like-minded throughout the development of the application. The requirements gathered were split into two groups because of the time restraint applied to the development process. The core required features have the basic functionality with the unique selling point of the application. Extended features were dependant on the remaining time. Low fidelity prototypes and diagrams were produced for the group to have a design to reference. This ensured sub-groups to be of like minds resulting in a reduction in conflicts.

It was required that the application was to be functioning on Google App Engine. As this is a new engine which many of the group did not have experience with, several meetings were initiated to ensure the understanding of the engine. A challenge was met as the group's experience relied on SQL relational databases, whereas Google App Engine applied hierarchical databases. The storage infrastructure used by GAE is that of a NoSQL document database, engineered by Google to focus on scaling, high performance and ease of development for the types of large-scale applications that run on the service. As a team we were more familiar with the more traditional relational database so trying to understand how the GAE datastore was built, used and queried took some time to adjust to at the start of the project. However, after then researching and learning about how the datastore operated we were able to define our different entities and their properties in the Python code as classes.

An additional, but minor, problem was the group invitations. When inviting users who do not have a Google email address into a group on the application, the recipient did not receive the email. However, there is full functionality when using Google email accounts to sign up and use the application.

A key design decision was how data was illustrated to the user. The tabbed view of Calendar, Tasks and Groups at the top of the page, allowed the group to decide to include a box with a small menu for the adding of new items to the calendar, a box listing deadlines, and a box containing colour keys to the free space on the right of the calendar.



Realisation of the complexity of the group feature that we had wanted to build into the calendar almost resulted in removing the feature altogether. However, we persevered as this feature improved the uniqueness of the application, and added a new level of functionality.

5 EVALUATION

At its final stage, the system has successfully fulfilled the core requirements of the original design. However, due to the time constraints a potential extension was not fulfilled. The group had the idea to include an automatic meeting allocation system. It would have worked by finding all the shared free slots in the calendars of the group members so that groups could easily plan events when everyone was free. Unfortunately, this idea was not realised, but can be noted as a good potential way of extending the functionality of the system.

The functionality of the system can be seen to meet all major requirements of an organisational application. As presented in section one, the features enable personal and group organisation with the system implemented. The unique selling point of the application is the group feature but as there are several organisational applications on the market, this is designed primarily for students.

Furthermore, throughout the process of implementing and developing the application, a core decision of how an invitation was sent for a user to join a group resulted in the choosing of a request email to the desired invitee. There was potential for the user experience to be improved by adding a notification system for invites instead, but time constraints did not allow implementation of this feature.

A potential improvement to the application would be to have data from the University of Southampton regarding modules to allow an improved user experience for students in this University. The data required would be the lecture timeslots and rooms, so it would not be necessary for the students to manually input their University schedule. On the other hand, the application could be integrated with the current timetable application SUSSED produces, removing the possibility of adding lecture slots that are not officially on the user's timetable.

The application is not secure from hackers who seek to gain control of the application or access user information. However, by implementing the Google Login system, the user must have a Google account in order to use the application. This prevents the database from storing personal data such as names and passwords. As the application is in Beta, attacks such as DDOS, SQL injection or parameter manipulation have not been tested, therefore creating an additional improvement to be made.