Project Specifications Document

A Mapped Information System of St John's College

Grade 12 PAT

Gordon Fleming

Contents

Project Specifications Document **1**
Summary: 3
Problem 3
Current Solution 3
Solution 3
Specifications of Program Function: 4
Login Screen 4
Main Screen 4
Dynamic Map 4
Query Form 5
Booking Form 5
User Details 6
Help Screen 6
Specifications of User Interface: 6
General Colour Scheme 6
Login Screen 7
Main Screen 7
User Details 7
Dynamic Map. 7
Query Form 8
Booking Form 8
Help Screen 8
Specification of Help: 8
Internal 8
External 9
Specifications of Data Storage: 9
Database 9
Fields data types 10
When are records created? 10
When are records accessed? 10
When are records updated? 11
Hardware and Software Requirements: 11
Hardware 11
Software 12

Summary:

Problem

The problem I plan to tackle with my application is considered to be a well known problem at St John's College. I find and have heard that many teachers struggle to identify which venues/classes are booked at what times and for what activities around the school premises, thus affecting productivity around the school as well as loss of class time… Venues end up being double booked or canceled last minute and no one is notified. The problem ranges from incorrect timings on timetables to mismanagement and misunderstanding between staff members, mainly with activities that aren't part of the standard timetable.

Current Solution

I am currently aware staff make use of a shared google sheets form for bookings and reservations in various venues around the school, but this is limiting and is not specifically designed for the task and is restricting its potential and functionality. By having an actual map, my program will make it very easy for staff to identify exactly what class is being used and for what by what staff, something a spreadsheet is unable to do.

Solution

I plan on creating a dynamic, interactive and functionally user friendly 2D map of St John's College which will allow staff and students to view the goings on around the school 24/7, everyday of the year. While interacting with the map one will be able to get a name and description of the venue, along with the current activity at that present time as well as be able to see upcoming events. The application will also allow user input for reservations and bookings of venues. This includes large venues from REA and sports fields to small maths classrooms and common rooms. The program will be developed as a web application making it very easy for anyone to access, as it will be running on the users web browser. The user will login through the google API, thus only granting users from St John's College access. The users, after login will be redirected to the main page, where the interactive map is situated.

Specifications of Program Function:

Login Screen

- The user will be directed to a login page to gain access
  - Here there will be buttons to use the google sign in and sign out, using the google API
  - When the sign in button is clicked, a separate window will open containing googles sign-in page.
  - Users then need to enter their email and password
  - Using google OAuth API, the login will not allow anyone who isn't under the St John's organisation (This will be set in googles admin options)
  - An error message will be issued if the user isn't under the set organisation, St John's College
  - Their profile image along with their name and surname will be shown on successful login
  - The user will then be automatically redirect to the home page, containing the map...

Main Screen

- This screen will encompass all the other components of the project
  - Dynamic map
  - Query form
  - Booking form
  - User details
  - Help
- The screen will focus on binding all these aspects in a functional and user friendly way by displaying the respective components with the correct spacing and in the correct ratio according to the users screen size.

Dynamic Map

- This is the core function of the whole application
  - Use the following to function
    - SVG's
    - JSP / HTML / CSS / Javascript / JSON
    - Java
    - SQL (SQLite)
    - Glassfish Server

- The map will have to change to the users specifications and therefore needs to take in user input through the
    - User checking the box for 1st floor, ground floor...
    - Querying tool
    - Booking tool
- The map will retrieve all its data about what's happening where and when by communicating with databases as described in 'Data Storage' containing all the relevant information
    - activities, venues, timings and the staff members all contained in their own respective tables
- The user will be able to dynamically interact with the map of St John's College, being able to
    - explore the various venues and facilities by hovering over the map
    - view the current activities in the various venues
    - see the name of the venue and details of what activity is happening at the users specified time in that venue
    - see the staff member who is currently in that classroom
- Allow users to book free slots for whatever activities the user specifies in the booking/reservation procedure, the venue will be shown as free if unoccupied
- Dynamically change the colour of the map according to what activity is taking place in the certain venue for that time

Query Form

The query form will allow the user to filter their searches. This will be done through a GUI interface (described in the GUI section). The users filter will then be processed, analysing the users input and transforming it into a SQL statement. The SQL will then return the users search results in the form of the dynamic 2D map.

- User can query the following, with their parameter using set, built-in functions
    - time
    - venue
    - activity
    - staff

Booking Form

This form will mostly be handled with Java and SQL. This form will process the users requests see if their request is valid within the database structure and fields and return a relevant message whether the

booking was successful or not. The booking form will insert information into the database, through a GUI with drop down menus to avoid conflicts and invalid inputs… This form is one of the main functions in the project. The form will, just like the query form, transform the user input into a SQL statement which will be handled in Java and passed through to the connected tblBooking table in the database where the SQL will be executed.

- The table for handling bookings
  - will have the users details inserted as a record
  - if invalid will insert no records
- Will have drop-downs for
  - Staff member
  - Venue
  - Week
  - Day
  - Period
  - Time
  - Date
  - Activity

User Details

The users details will contain the login information of the user from google, here they will be able to sign out and see their details, including their full name and profile image. The main functionality here is handled by google's API. The sign out button will also be placed here and will reload and redirect the user when toggled.

Help Screen

The help button when 'clicked on' will simply open an image stored on the server and will display the mainpage with all of its features functions described in a straightforward fashion. If the user should require any other help or experiences a bug they will be able to contact the developer for further assistance. Tooltips will also be functioning on the page when user hovers over a tool will describe in-short its functionality.

Specifications of User Interface:

General Colour Scheme

I will maintain a general dark theme in my project, though the map will have various colours identifying the numerous classrooms and venues

around the school. The background colour will be dark with white and black text contrasting the general dark / grey theme. The different functionality laid out around the map will be 'lighter' contrasting colours.

Login Screen

The login screen will contain a simple google "Sign in" button, centered with a dark background and title of the project above. The button will open in a different window with the generic google login interface. I will customise the button for logout, matching it with the main google sign in button. There will be a small 'Help' hyperlink in the bottom right corner.

Main Screen

The main screen will have a dark background with the title of the application centred at the top of the users screen. The details of who made the program and current possible version will be displayed in the bottom center of the users screen. The map will be centered in the middle of the users screen and scale down or up depending on the screen size of the user. The users details will be in either of the top corners and other map functionality will be appropriately spaced for easy to use, simple and effective appeal. The query and booking tools placed above the map and to the right side respectively. There will be a small 'Help' hyperlink in the bottom right corner.

User Details

The users details will be displayed in either the top right or top left corners of the users screen. The users details will contain their profile image used for their school google account along with their full name. This information all neatly displayed under each other.

Dynamic Map.

The map will take the majority of the screen real estate, the map will scale depending on the screen size of the user to produce an effective viewing scale, large enough to make out the detail but small enough to encompass an understanding of location. The numerous venues will all be separately coloured for easy identification between activities in the separate venues. The map will have a check box below to handle the different floors of St John's College. The current activities will be displayed in their respective locations. Lastly the UI will include an input block, to allow users to book/reserve venues for specific times and dates.

Query Form

The query form will be above the map with drop down menus to query various fields in different ways. This will be a bar containing a drop down section of the field the user is querying, how they want to go about querying that field and finally a box with the users parameters.

Booking Form

The booking form will be in either of the remaining sides of the screen. This will be a set form with drop down menus as well as blank text boxes and a 'Submit' button, with a message saying whether the booking is successful or not when processed. The booking form will have descriptions above each drop down and a description box where the user can enter their miscellaneous booking details. The booking form will try to maintain the same look and appearance as the query form.

Help Screen

Is a small button (hyperlink) coloured in yellow, which will open an image. The image will be an effectively stylised screenshot of the main screen with text describing all the different functions in their respective locations around the main page. The image will have arrows pointing to the tools and different functions with relevant descriptions attached. This will be placed in the bottom right for both login and main screens.

Specification of Help:

Internal

The help will be provided through either contact with the developer through email or/and by 'clicking on' a button which will display an image detailing the main page as described in the UI description. This should provide the user with enough information to solve their issues. I will also add tooltips, so as the user hovers over functions, help will pop up.

- Login Screen
    - Help button which when toggled will display an image of how to login and will describe the
        - Sign in button
        - Sign out button
        - Google sign in (inform the user to use their St John's details for a successful login)

        - Inform the user what will happen when successful
- Main Page
    - Tooltips
        - Provide short descriptions of all the functions on what they are used for
        - The tooltips will pop-up when user hovers over the function with their mouse pointer
    - Help button (hyperlink)
        - When toggled will display an edited screenshot of the main screen with arrows and descriptions
        - The image will have detailed descriptions of all the functions and other miscellaneous items
            - Describe in the main page how to
                - Query / search the dynamic map using the specified query tool
                - Book using the booking form
                - Sign out
                - Contact for more help
                - Interact with the map
                    - zoom in and out
                    - explain the maps symbols
                    - hint the user to refer to the map 'Legend' to understand the map more

External

The external help will comprise of a simple README.txt file containing

- User instructions
- General information
- Additional help
- Github repository link
- Open source license details

Specifications of Data Storage:

Database

I will be using a SQLite database with the Spatialite extension as it's simple, lightweight and widely accepted (it's also free and open source). The database will contain several tables each containing different information, from staff members to venues to timetables and activities.

The datatypes of all stored data will be relevant to their use case and in standard formats. This is a broad overview of some table designs.

tblTimings (TimeID, PeriodName, StartTime, EndTime, WeekDay, Week)
tblTimeTables(StaffID, Week, Weekday, Period_0, Period_1, Period_2, Period_3, Period_4, Period_5, Period_6, Period_7, Period_8, Period_9)
tblVenues (VenueID, Name, Short, Geometry)
tblStaffMembers (StaffID, StaffName, Initials)
tblActivities (ActivityID, ActivityName, Short)
tblBooking (TimeID, VenueID, StaffID, ActivityID, Date)

Fields data types

- Auto_Increment
  - VenueID
  - ActivityID
  - StaffID
- Composite_Primary_Keys
  - TimeID
  - StaffID
  - DefaultActivityID
- String
  - ActivityName
  - Date (In correct format)
  - Week
  - WeekDay
  - PeriodName
  - StartTime
  - EndTime
  - StaffName
  - Period_0 to Period_9
- Geometry (Geospatial)
  - Geometry

When are records created?

- Users input their booking details (tblBooking)
- New staff members (tblStaffMembers)
- New rooms and facilities (tblVenue)
- New subjects and activities

When are records accessed?

- Querying for a certain venue, subject, teacher or time...
- Details are requested from the user about a specific venue
- The users interact with the map
- tblVenue accessed constantly to keep the map up-to-date

When are records updated?

- Venues change their use case or / and name
- Details need to be changed for already booked venues
- New timetable with different times or / and periods
- Staff's timetables change

These tables will be used to render and populate the map with information, bookings and other data. Records will be created in tblBooking whenever a successful user request has been processed. The other tables will have their data pre-populated and occasionally updated when necessary.

Hardware and Software Requirements:

Hardware

The hardware for development and deployment is minimal as the program will be a relatively lightweight application and therefore easy to manage as well as not demanding. The processing will be server-side, thus making the experience for the user more fluid and making the experience as seamless as possible. The application will require a server to run 24/7, powerful enough to handle the processing the users requests. The server will store the database along with all the applications assets and functioning code.

Deployment Minimum Requirements

- Intel Core i3 @3.7GHz
- 8GB RAM
- 200GB HDD
- 450W PSU
- Compatible MOBO

Users Requirements

Minimum hardware requirements for the specified browsers:

	Chrome	Opera	Safari
Processor Windows	Pentium 4	Pentium II	500-MHz Pentium-class
Process Mac	Intel	Intel	
Min RAM	128 MB	128 MB	256 MB
Recommended RAM		256 MB	
Min Disk Space	100 MB	20 MB	unknown
Rec. Disk Space		100 MB	
Rec. Disk Space 64-bit			
Windows	Windows XP SP2	Windows 2000	Windows XP SP2
OS X	OS X 10.5.6	Mac OS X 10.5	OS X 10.5.8
Linux	Ubuntu 10.04 ...	Any recent	Not available

Software

- Glassfish server 4.1.1
- Netbeans 8.2 (JavaEE version)
- JDK 1.8
- Any operating system that can run a compatible browser, eg: Windows 10
- Notepad ++ 7.6
- Chrome, Safari, Firefox and Opera…
- DB Browser for SQLite (Version 3.11.1)
- GitHub Desktop (Version 1.6.3)