Deliverable 2: Software Architecture

The University of New South Wales

SENG2021 2019 T1

BEAMS

MOZAMEL ANWARY	Z5209948
BEI CHEN	Z5117096
PAUL GRACE	Z5208656
SUMAYYAH HO	Z5210234
MARIYA SHMALKO	Z5209377

Table of Contents

Introduction

This report is the second phase of our project development. The aim is to address, in detail, the aspects of the software architecture that will be implemented in our design of the application BEAMS, as well as the incentive behind each decision. This includes:
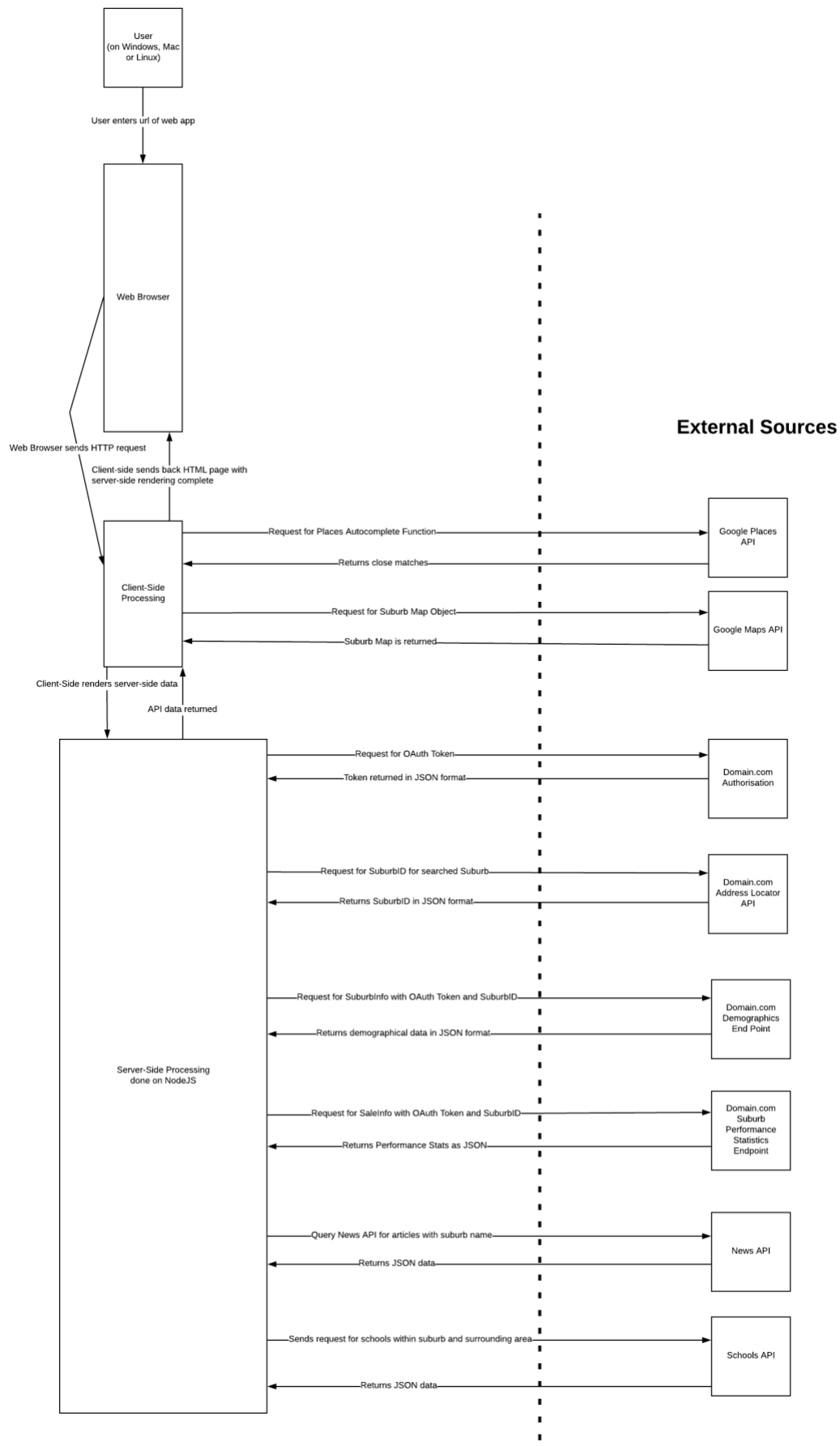
- External data sources
- Software components
  - Web Stack
    - Third party components
    - Developing components
- Language implemented
- Framework
- Platform

Software Architecture

External Data Sources

BEAMS will require access to APIs related to geographic and demographic data, which will be employed in representing relevant data across different mediums. The system will involve the APIs of Google places, Google maps, and Geocoder API to allow users to search for and view a suburb on a map as well as those surrounding it with ease. Data associated with suburb statistics including demographics, and infrastructure will be obtained from Domain.com's end-points labelled: Address Locators, Demographics, Schools and Suburb Performance Statistics. BEAMS will be using the News API from newsapi.org. News API will mainly be used to provide the last six months' worth of top headline news within the country, and suburb, of choice. The News API will collect these breaking news headlines and articles from over 30,000 news sources and blogs. Information about schools will be obtained from both HERE's geocoder API and Domain.com's GET Schools location end-point. The API provides access to suburb coordinates (longitude and latitude), which will then be used with Domain's Get Schools location end-point to find schools in close proximity of the given suburb.

Software components

User
(on Windows, Mac
or Linux)

User enters url of web app

Web Browser

External Sources

Web Browser sends HTTP request

Client-side sends back HTML page with
server-side rendering complete

Client-Side
Processing

Request for Places Autocomplete Function

Returns close matches

Google Places
API

Request for Suburb Map Object

Suburb Map is returned

Google Maps API

Client-Side renders server-side data

API data returned

Request for OAuth Token

Token returned in JSON format

Domain.com
Authorisation

Request for SuburbID for searched Suburb

Returns SuburbID in JSON format

Domain.com
Address Locator
API

Request for SuburbInfo with OAuth Token and SuburbID

Returns demographical data in JSON format

Domain.com
Demographics
End Point

Server-Side Processing
done on NodeJS

Request for SaleInfo with OAuth Token and SuburbID

Returns Performance Stats as JSON

Domain.com
Suburb
Performance
Statistics
Endpoint

Query News API for articles with suburb name

Returns JSON data

News API

Sends request for schools within suburb and surrounding area

Returns JSON data

Schools API

Language Implementation of software components

In terms of languages, BEAMS will be implementing the server-side on NodeJS and ExpressJS. Client side will be rendered using HTML, CSS and JavaScript with JSX utilised to syntactically enhance the Web App.

Implementation technology/framework

Development

BEAMS's client-side will be implemented through React, a JavaScript library that allows developers to create interactive user interfaces for web applications. The React infrastructure is comprised of reusable, composable, and stateful components, allowing for seamless integration and modification of the codebase to be used in the BEAMS application. React updates also results in rapid component reload times and dynamic state changes when components are altered. This allows developers to view changes in components without the need to reload the application, resulting in greater time efficiency; this could benefit BEAMS particularly due to the limited time frame of the project. Acknowledging that the development team has minimal experience with frameworks for web-apps, it would be an appropriate decision to employ React to the project due to its shallow learning curve. There is also an abundance of supporting resources to utilise with this framework, including detailed documentation, a range of easy-to-use, accessible online libraries, and forums which include resolutions to common roadblocks.

The server-side for this project will be implemented through Node.js and Express.js. Node.js is a Javascript run-time environment that allows BEAMS to securely initiate communication with API end-points. Express.js is a node.js web application framework that provides a set of flexible and easy-to-use features. The benefit of these two technologies is that they provide a simple server to BEAMS that allows communication to end-points securely without revealing client secrets and client-id.

Deployment

While the team has not yet deployed the web app, BEAMS has decided to host on AWS. This is due to the fact that some are familiar with the technology.

Platform

BEAMS will be deployed as a web application, with the final system intended to be compatible for desktop devices that support HTML web browsers; this includes Windows, Linux and MacOS systems. This will support the vast majority of the potential user-base on PC devices, with it being available using browsers including Chrome (for Windows, MacOS) and Chromium (for Linux). The application will not be tailored to mobile devices due to its lack of utility; minimal screen estate will make it inappropriate to use certain features, such as comparing detail between suburbs or viewing multiple data sets (i.e. graphs, maps, etc.) simultaneously. The application will be hosted via AWS.

Key benefits/achievements of architectural choices

BEAMS architectural choices have a few key benefits including the shallow learning curve, flexibility and security. The end goal of BEAMS is to provide users with access to an interface that allows them to easily browse and understand the real estate market with reference to specific suburbs. The architectural choices made in preparation for this project all aid in achieving that goal as React, Node.js, and Express.js allow cohesive implementation of functionality and key features. The decisions to use HTML, CSS, and JavaScript with JSX provide greater enhancement of the Web App for the client while maintaining a clean and simple and intuitive interface that is consistent with the user stories of this project.

Initial Software Design
Updated requirements and user stories

Requirement 1: User is able to search areas by suburb and refine search by property criteria.

Feature 1.1: User can search for suburb by name.

US ID	1.1.0
Priority	Critical for MVP
Estimate	3 SP
User Story **As a** user, **I want to** search for suburbs by their name **so that** I can find the most relevant search results for me.	
Given that I am at the search webpage **When** I type in a suburb name. **Then** I can view my preferred suburb(s).	

Feature 1.2: User can filter search depending on property criteria.

US ID	1.2.0
Priority	Critical for MVP
Estimate	2 SP
User Story **As a** user, **I want to** select the property type which is most suitable for me **so that** I can narrow my search and find information on properties most relevant to my preferences.	
Given that I have searched for a suburb by name. **When** I select my preferred property type. **Then** I am shown a list of suburbs with my filters remembered. **When** I select my preferred suburb, **Then** I am shown my preferred suburb and statistics about those property types.	

US ID	1.2.1
Priority	Critical for MVP
Estimate	2 SP

User Story
As a user, **I want to** select the price range which is most suitable for me **so that** I can refine my search and find information on properties most appropriate to my preferences.

Given that I have searched for a suburb by name.
When I select my preferred price.
Then I am shown a list of suburbs with my filters remembered.
When I select my preferred suburb,
Then I am shown my preferred suburb and statistics in the price range.

US ID	1.2.2
Priority	Critical for MVP
Estimate	2 SP

User Story
As a user, **I want to** select the number of bedrooms a property has **so that** I can refine my search and find information on properties most appropriate to my preferences.

Given that I have searched for a suburb by name.
When I select my preferred number of bedrooms,
Then I am shown a list of suburbs with my filters remembered.
When I select my preferred suburb,
Then I am shown my preferred suburb and statistics about properties with those many bedrooms.

Requirement 2: There should be a clear graphical representation for housing market data.

Feature 2.1: User can view graphical representation of suburb's housing market trends after searching for the suburb.

US ID	2.1.0
Priority	Critical for MVP
Estimate	2 SP

User Story
As a user, **I want** an easy to read, graphical representation of the average property price in the suburb I am researching **so that** I can identify trends and improve my decisions.
Given that I have searched for my preferred suburb(s)   **When** I click on my preferred suburb,   **Then** I am shown a graphical representation of the average property price in the suburb.

US ID	2.1.1
Priority	Critical for MVP
Estimate	2 SP

User Story
As a user, **I want** graphical representation of the average property selling price or renting price in the suburb I am researching according to the filters I have selected **so that** I can filter the information according to my preferences.
Given that I have searched for my preferred suburb(s)   **When** I click on my preferred suburb,   **Then** I am shown a graphical representation of the average property, filtered by my criteria, price in the suburb.

Requirement 3: There should be a suburb profile including demographic, infrastructure, schools and public transport information.

Feature 3.1: User can access demographic data.

US ID	3.1.0
Priority	Critical for MVP
Estimate	2 SP
User Story **As a** user, **I want** to access demographic data, such as crime rate, family type, median age and average income, on my preferred suburb(s) **so that** I can find the best property most appropriate to my lifestyle.	
Given that I am viewing a suburb profile page **When** I click on the demographic data tab **Then** I am shown an easy to understand representation of demographic data in the suburb.	

Feature 3.2: User can access infrastructure data.

US ID	3.2.0
Priority	High
Estimate	2 SP
User Story **As a** user, **I want** to access infrastructure data, such as medical facilities, shopping centres and parks, on my preferred suburb(s) **so that** I can find the best suburb most appropriate to my lifestyle.	
Given that I am viewing a suburb profile page **When** I click on the infrastructure data tab **Then** I am shown an easy to understand representation of infrastructure data in the suburb.	

Feature 3.3: User can access schools within the suburb and surrounding area and their data.

US ID	3.3.0
Priority	Critical for MVP
Estimate	2 SP
User Story **As a** user, **I want** to view schools within my preferred suburb(s) **so that** I can find the best suburb most appropriate to my lifestyle.	
Given that I am viewing a suburb profile page **When** I click on the education data tab **Then** I am shown an easy to understand representation of education data in the suburb.	

Feature 3.4: User can access data about available public transport in the area.

US ID	3.4.0
Priority	High
Estimate	2 SP
User Story **As a** user, **I want** to access data about public transport in my preferred suburb(s) **so that** I can find the best suburb most appropriate to my lifestyle.	
Given that I am viewing a suburb profile page **When** I click on the public transport data tab **Then** I am shown an easy to understand representation of different public transport options in the suburb.	

Requirement 4: News filtered by suburb is visible on the suburb page.

Feature 4.1: User can view the news relevant to a suburb on the suburb page.

US ID	4.1.0
Priority	Medium
Estimate	1 SP
User Story **As a** user, **I want** to view recent news headlines relevant to a suburb **so that** I can be aware of the local community's culture, achievements and/or hazards.	
Given that I have searched for my preferred suburb(s) **When** I click on the suburb's profile page. **Then** I am shown recent news headlines specific to the suburb and surrounding area.	

US ID	4.1.1
Priority	3
Estimate	1 SP
User Story **As a** user, **I want** to expand recent news relevant to a suburb **so that** I can further my understanding of recent events within the region.	
Given that I am on the suburb's profile, **When** I click on the news headline that interests me, **Then** I am redirected to the headline's source website.	

Requirement 5: Website should suggest optimal time, based off of real estate market trends and predictions, to purchase in the area with the given criteria.

Feature 5.1: User can view real estate market trends that indicate the most optimal time to purchase a property.

US ID	5.1.0
Priority	High
Estimate	4 SP
User Story **As a** user, **I want** to be given an accurate estimate as to when to buy a property in a given suburb **so that** I can make a financially mature decision.	
Given that I am viewing a suburb profile page **When** I view the house market data. **Then** I am shown the optimal time to purchase a property in the preferred suburb.	

Requirement 6: Website should show suburb area on a map.
Feature 6.1: User can view suburb area on a map

US ID	6.1.0
Priority	High
Estimate	3 SP
User Story **As a** user, **I want** to view the suburb area on a map **so that** I can have a visual idea of the area geographically.	
Given that I have searched for my preferred suburb(s) **When** I click on the suburb's profile page. **Then** I am shown a map with the suburb's area outlined.	

Sequence Diagram

| Browser | Application | Server | Domain API | News API | Places API | Maps Embedded API |

1.1.0
As a user, I want to search for suburbs by their name so that I can find the most relevant search results for me.

See 3.1.0 – 3.1.4 for displaying suburb information

- HomePage.js
- enterSuburb()
- getSuburbName()
- SuburbInfo().json
- HomePage.js

1.2.0 -1.2.2
As a user, I want to search for a particular type of property in a given suburb within a given price range

- submitQuery()
- getSuburbInfo()
- getToken()
- Token.txt
- getSuburbId(token, suburb,state)
- SuburbId.json
- getPropertyPrices(token, property, bedrooms)
- PropertyPrices.json
- SuburbInfo().json
- SuburbPage.js

2.1.0 - 2.1.1
As a user, I want an easy to read, graphical representation of the average property price in the suburb I am researching so that I can identify trends and improve my decisions.

- submitQuery()
- getSuburbInfo()
- getSuburbId(token, suburb,state)
- SuburbId.json
- getPropertyPrices(token, property, bedrooms)
- PropertyPrices.json
- SuburbInfo().json
- SuburbPage.js

3.1.0 - 3.4.0
As a user, I want to view/access

- demographic data, such as crime rate, family type, median age and average income,
- infrastructure data, such as medical facilities, shopping centres and
- Parks
- schools within my preferred suburb
- data about public transport

in my preferred suburb(s) so that I can find the best property most appropriate to my lifestyle.

4.1.0 - 4.1.1
As a user, I want to view and expand recent news headlines relevant to a suburb so that I can be aware of
the local community's culture, achievements and/ or hazards.

6.1.0
As a user, I want to view the suburb area on a map so that I can have a visual idea of the area geographically.

- submitQuery()
- getSuburbInfo()
- getToken()
- Token
- getSuburbId(token, suburb, state)
- SuburbId.json
- getDemographics(token, suburb)
- Demographics.json
- getNews(token, suburb)
- News
- SuburbInfo.json
- getMap(suburb)
- Map
- SuburbPage.js
- expandNews()
- SuburbPage.js

5.1.0
As a user, I want to be given an accurate estimate as to when to buy a property in a given
suburb so that I can make a financially mature decision.

- showTrends()
- getSuburbInfo()
- getTrends(token, suburb, property, bedrooms)
- Trends.json
- getPropertyPrices(token, property, bedrooms)
- PropertyPrices.json
- SuburbInfo.json
- SuburbPage.js