COMP3900-H15A-capSquad - Project Proposal

Daniel Latimer z5115175
Connor O'Shea z5115177
Kevin Chan z5113136
Oliver Richards z5157383
Peter Kerr z5115807

September 30, 2020

Contents

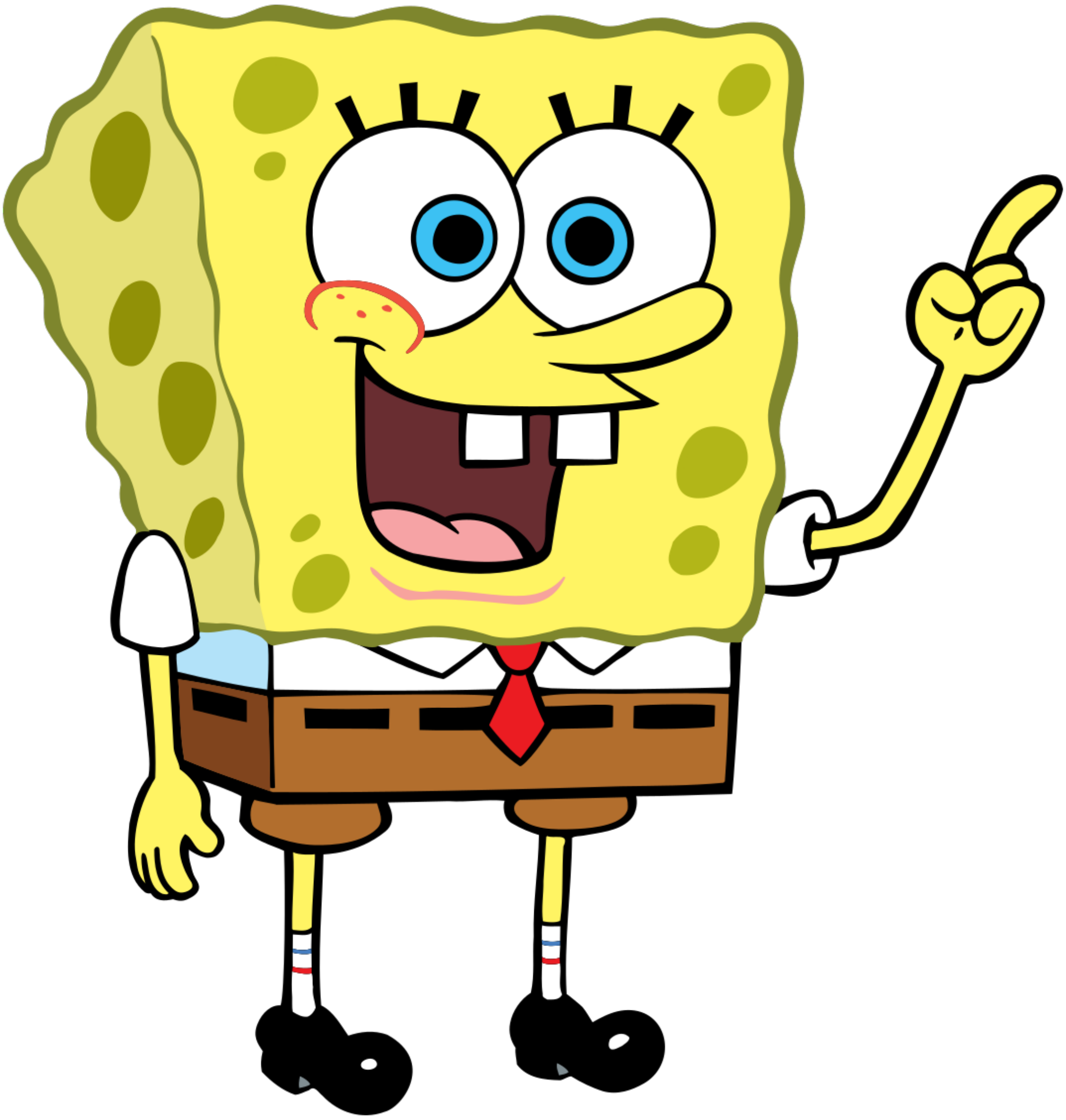


Figure 1: Background? [1]

1 Background

delete me

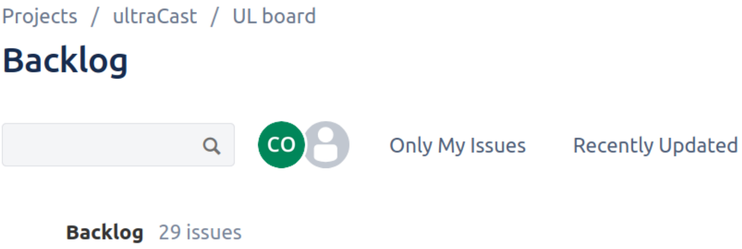
1.1 Problem Domain

1.2 Existing Systems

2 User Stories

2.1 Product Backlog

The 29 user stories which make up the product backlog, as shown in Figure XXX, were grouped into three categories as described in the sections 2.1.1-3. Screenshots of the stories which make up the backlog can be found in sections 2.1.1-3.



Projects / ultraCast / UL board

Backlog

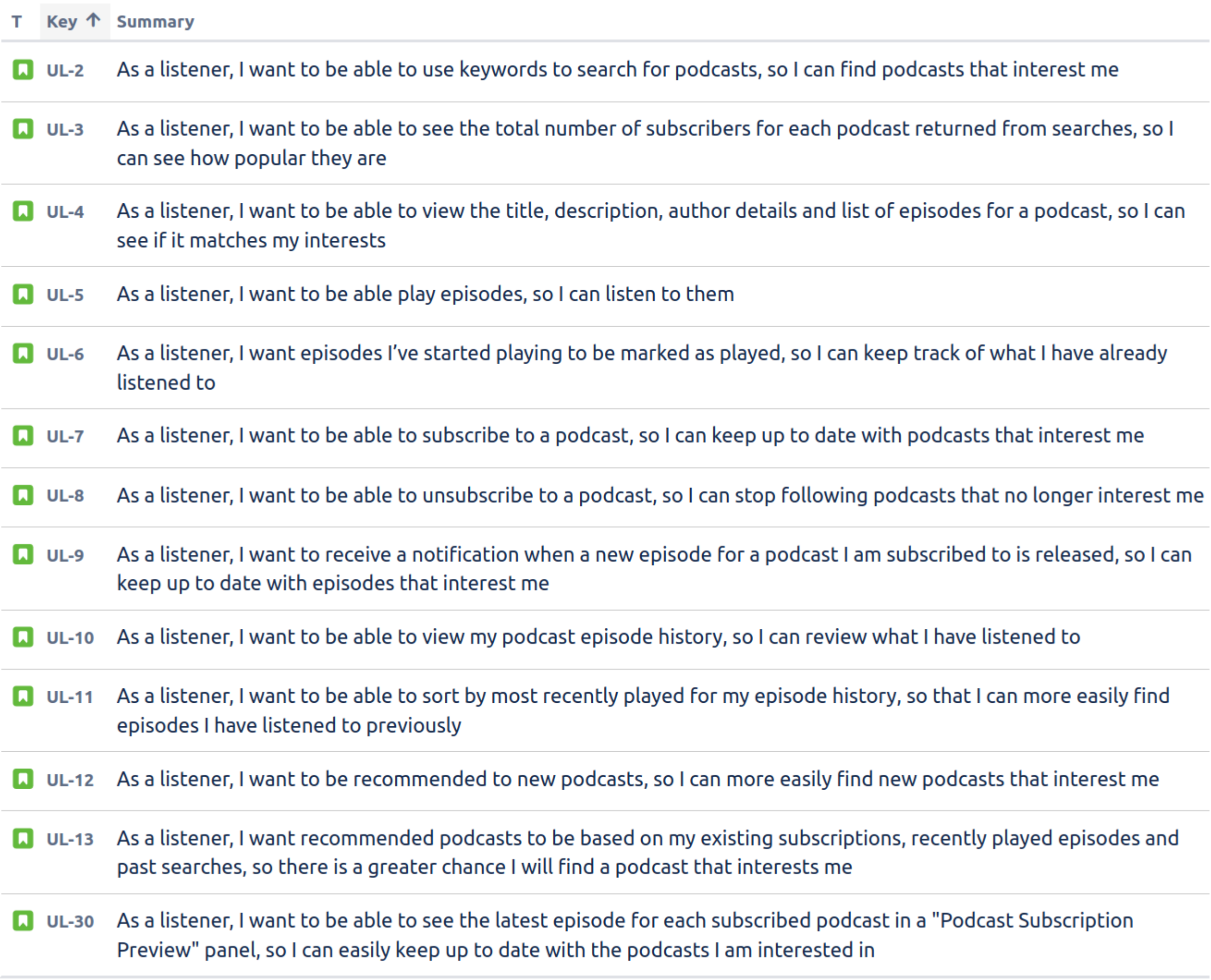
CO Only My Issues Recently Updated

Backlog 29 issues

Figure 2: ultraCast Backlog Count

2.1.1 Project Objectives Stories

The project objective stories were derived directly from the project objectives. These JIRA stories can be found in Figure XXX below.



T	Key ↑	Summary
🟩	UL-2	As a listener, I want to be able to use keywords to search for podcasts, so I can find podcasts that interest me
🟩	UL-3	As a listener, I want to be able to see the total number of subscribers for each podcast returned from searches, so I can see how popular they are
🟩	UL-4	As a listener, I want to be able to view the title, description, author details and list of episodes for a podcast, so I can see if it matches my interests
🟩	UL-5	As a listener, I want to be able play episodes, so I can listen to them
🟩	UL-6	As a listener, I want episodes I've started playing to be marked as played, so I can keep track of what I have already listened to
🟩	UL-7	As a listener, I want to be able to subscribe to a podcast, so I can keep up to date with podcasts that interest me
🟩	UL-8	As a listener, I want to be able to unsubscribe to a podcast, so I can stop following podcasts that no longer interest me
🟩	UL-9	As a listener, I want to receive a notification when a new episode for a podcast I am subscribed to is released, so I can keep up to date with episodes that interest me
🟩	UL-10	As a listener, I want to be able to view my podcast episode history, so I can review what I have listened to
🟩	UL-11	As a listener, I want to be able to sort by most recently played for my episode history, so that I can more easily find episodes I have listened to previously
🟩	UL-12	As a listener, I want to be recommended to new podcasts, so I can more easily find new podcasts that interest me
🟩	UL-13	As a listener, I want recommended podcasts to be based on my existing subscriptions, recently played episodes and past searches, so there is a greater chance I will find a podcast that interests me
🟩	UL-30	As a listener, I want to be able to see the latest episode for each subscribed podcast in a "Podcast Subscription Preview" panel, so I can easily keep up to date with the podcasts I am interested in

Figure 3: JIRA Objective User Stories

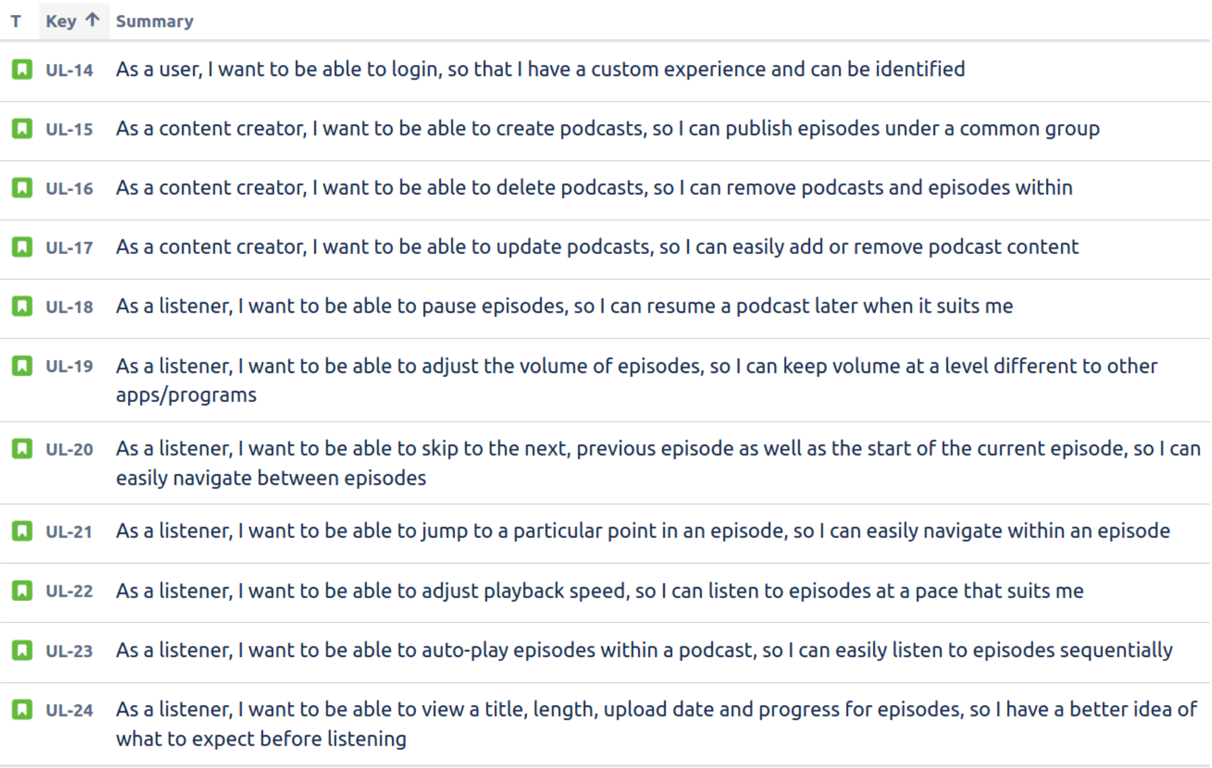
The mapping of each project objective to the final user story was summarised in Table XXX.

2.1.2 System Stories

The system stories were designed to address common features offered by existing offerings in the same problem domain. They can be seen in Figure XXX below.

Table 1: Project Objectives to Stories Mapping

Project Objective	**Story Key**
Listeners must be able to search for podcasts that interest them by keywords, resulting in a list of matching podcast titles, where the total number of subscriptions on the ultraCast platform (function described later) for each podcast is shown next to the title	UL-2 UL-3
Listeners must be able to select a podcast show from returned search results to view its full details, including its title, description, any author details that exist, as well as a list of episodes for the show	UL-4
Listeners must be able to play a selected episode within a podcast show, and once that episode starts being played, the listener must be able to also clearly see this episode marked as ”Played”	UL-5 UL-6
Listeners must be able to subscribe or unsubscribe from a podcast show	UL-7 UL-8
Listeners must be able to see the latest episode available for each show that they subscribed to in a ”Podcast Subscription Preview” panel	UL-30
Listeners must be notified by the platform when a new episode for a show they are subscribed appears	UL-9
Listeners must be able to see a history of the podcast episodes that they have played, sorted in order from most recently played to least recently played	UL-10 UL-11
ultraCast must be able to recommend new podcast shows to a listener based on at least information about the podcast shows they are subscribed to, podcast episodes they have recently played, and their past podcast searches	UL-12 UL-13

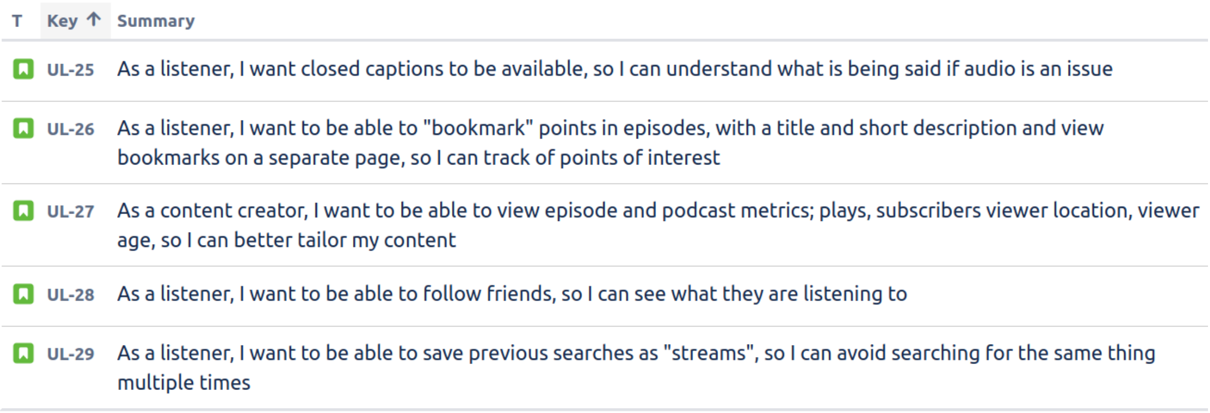


T	Key ↑	Summary
🔖	UL-14	As a user, I want to be able to login, so that I have a custom experience and can be identified
🔖	UL-15	As a content creator, I want to be able to create podcasts, so I can publish episodes under a common group
🔖	UL-16	As a content creator, I want to be able to delete podcasts, so I can remove podcasts and episodes within
🔖	UL-17	As a content creator, I want to be able to update podcasts, so I can easily add or remove podcast content
🔖	UL-18	As a listener, I want to be able to pause episodes, so I can resume a podcast later when it suits me
🔖	UL-19	As a listener, I want to be able to adjust the volume of episodes, so I can keep volume at a level different to other apps/programs
🔖	UL-20	As a listener, I want to be able to skip to the next, previous episode as well as the start of the current episode, so I can easily navigate between episodes
🔖	UL-21	As a listener, I want to be able to jump to a particular point in an episode, so I can easily navigate within an episode
🔖	UL-22	As a listener, I want to be able to adjust playback speed, so I can listen to episodes at a pace that suits me
🔖	UL-23	As a listener, I want to be able to auto-play episodes within a podcast, so I can easily listen to episodes sequentially
🔖	UL-24	As a listener, I want to be able to view a title, length, upload date and progress for episodes, so I have a better idea of what to expect before listening

Figure 4: JIRA System User Stories

2.1.3 Novel Features Stories

The novel feature user stories, as shown in Figure XXX, were designed to create desirable features that are either uncommon or not available in other mainstream offerings in the same problem domain.



T	Key ↑	Summary
🔖	UL-25	As a listener, I want closed captions to be available, so I can understand what is being said if audio is an issue
🔖	UL-26	As a listener, I want to be able to "bookmark" points in episodes, with a title and short description and view bookmarks on a separate page, so I can track of points of interest
🔖	UL-27	As a content creator, I want to be able to view episode and podcast metrics; plays, subscribers viewer location, viewer age, so I can better tailor my content
🔖	UL-28	As a listener, I want to be able to follow friends, so I can see what they are listening to
🔖	UL-29	As a listener, I want to be able to save previous searches as "streams", so I can avoid searching for the same thing multiple times

Figure 5: JIRA Novel User Stories

TODO format as table using competitiors and user story numbers in figure above

2.2 Sprints

3 Interface and Flow Diagrams

4 System Architecture

The proposed system architecture can be seen in Figure 6. It can be seen that our end users will be podcast listeners and content creators. First, we will use MongoDB to store our data, a NoSQL database that is popular for its high scalability. This service will be interfaced with the MongoDB-Python driver, available on the MongoDB website. Next, we will be using Flask for our web-server: a micro-framework that allows us to quickly develop an MVP solution. Flask is written in Python, so connecting to the database via the MongoDB-Python driver should be straightforward. Additionally, we will have a recommendation service that will generate recommended podcasts based on the users listening history. Finally, we will have a React frontend application that will enable our users to login, search and play podcasts, and get recommendations on ones they may be interested in. The React application and Flask application will communicate through a GraphQL API: a scalable alternative to the popular REST API.

The architecture has been designed with the final demonstration in mind, hence, the business and presentation layers are shown to be hosted on the VLab machine. Currently, MongoDB is not supported by Debian 6 (the Linux environment on the VLab machine), so we have opted to put the data layer onto an AWS EC2 instance.

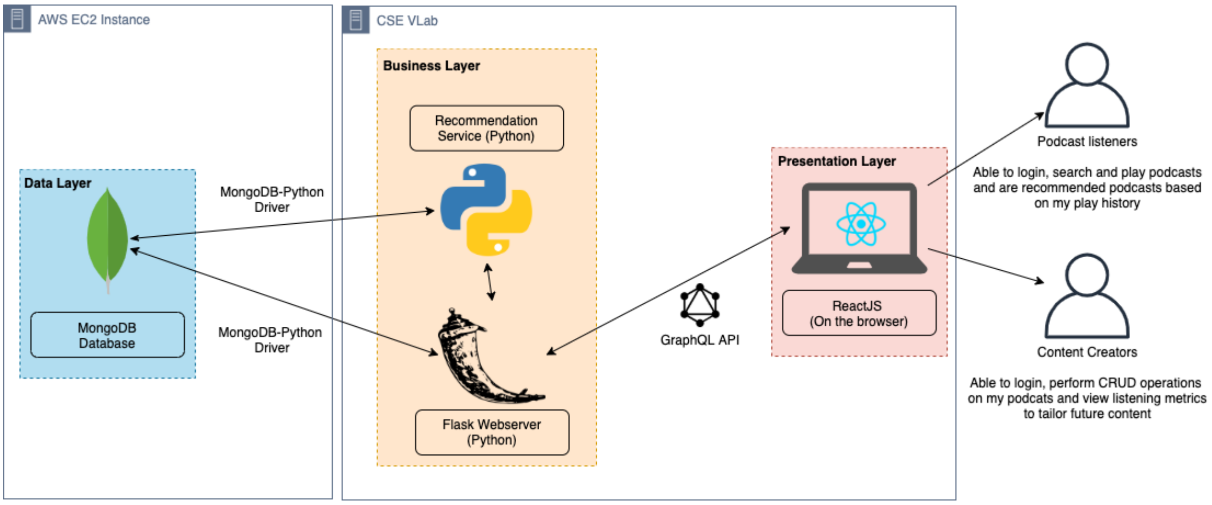

AWS EC2 Instance
CSE VLab
Business Layer
Recommendation Service (Python)
Data Layer
MongoDB-Python Driver
MongoDB Database
MongoDB-Python Driver
Flask Webserver (Python)
GraphQL API
Presentation Layer
ReactJS (On the browser)
Podcast listeners
Able to login, search and play podcasts and are recommended podcasts based on my play history
Content Creators
Able to login, perform CRUD operations on my podcats and view listening metrics to tailor future content

Figure 6: Proposed System Architecture

References

[1] John E. Laird, Keegan R. Kinkade, Shiwali Mohan, and Joseph Z. Xu. Cognitive robotics using the soar cognitive architecture. *AAAI Workshop - Technical Report*, WS-12-06:46–54, 2012.