Amplify Software Development Plan

Alexia Filler, Ben Kern, Ian Lizarda, Donovan Moini and Serena Zafiris

Table of Contents

4.1 Plan Introduction

This Software Development Plan provides the details of the planned development for the Amplify Software CSCI which provides an application to allow registered users to create and join public playlist rooms. Amplify will be a community music streaming service that will allow users to create and join rooms that host public music playlists using Spotify and YouTube services, similar to platforms like plug.dj and Outloud. Users will be able to join rooms and add songs of their choice from Spotify or YouTube to the community queue, which will be cycled through in round-robin style between all users' queued songs. Amplify will also have features useful to an individual user; the user will be able to import multiple playlists and rotate through those playlists via round-robin style. The frontend will use React and Typescript. The backend will use JavaScript, Apollo GraphQL, Amazon Cognito, Amazon Relational Database Services, Amazon Gateway, and Amazon Lambda.

4.1.1 Project Deliverables

4.1.1.1 Project Proposal Presentation

- Week 01
- Provide the instructor and class with an informative presentation covering the conceptual overview, tools, elements, and projected timeline of the Amplify project with the goal of both informing the audience about the project, but also of recruiting one to two more members to the project.

4.1.1.2 Project Proposal Document

- Due Week 02
- Provide the instructor with a document containing necessary information about the proposed Amplify project (tools to be used, team division, general outline and goal of the project, etc.). This document will contain both the description and the justification of the project.

4.1.1.3 Requirements Specification

- Due Week 02
- The Software Requirements Specification is to be turned in to the instructor and contain details about the low-level requirements of the Amplify project. Specifically, this document will contain an introduction, the breakdown of the CSCI

components, functional requirements, performance requirements, and requirements for the project, development, and execution environments. It outlines the gritty details of the contract of the functionality of the application agreed upon between developer (student) and client (instructor).

4.1.1.4 Initial Development Schedule (Part of SDP draft)

- Due Week 05
- This deliverable will provide an outline of the expected progress of the project over the given timeline (16 weeks), including deadlines and anticipated completion dates/checkpoints.

4.1.1.5 Oral Status Report

- Week 05
- Oral status reports will give the team an opportunity to go over what has been done thus far and what still needs to be done, either in the current sprint or farther along in the schedule. It will also allow the team to update the client (instructor) on the current progress of the project.

4.1.1.6 Software Development Plan Document

- Due Week 07
- This document will detail the software development process that will be used throughout the production of the Amplify project, including details about the project deliverables, software and hardware resources, organization, and schedule.

4.1.1.7 Project Preliminary Design Review Presentation

- Due Week 11
- This presentation will act as an update to the class and instructor about the progress of the design of the Amplify project. It will allow for preliminary criticism on the project design so that changes can be made, if necessary.

4.1.1.8 Software Design Description Document (Architecture Section)

- Due Week 11
- This document will contain current details on the planned implementation of the functionality specified in the Software Requirements

Specification. This will require careful analysis of the project design and how the different elements interact and fit together.

4.1.1.9 Project Critical Design Review Presentation

- Due Week 12
- This presentation will act as an update to the class and instructor about the progress of the design of the Amplify project. It will allow for criticism on the project design so that vital changes can be made, if necessary.

4.1.1.10 Software Design Description

- Due Week 12
- This document will detail the overall architecture of Amplify's software system, as well as including specifications on the system's design. It will provide a high-level description of the project's various interfaces, including hardware, software, and human interfaces, as well as details about the classes, interfaces, and data structures used within the project. Many diagrams will be used to provide a better explanation of how the different elements of the project tie together.

4.1.1.11 ALPHA/BETA Presentation/Demonstration

- Due Week 14
- This presentation will act as an update to the class and instructor about the progress of the design of the Amplify project. It will provide a demonstration of the application's uses in its Alpha or Beta level (nearing the final delivery of the completed application).

4.1.1.12 Test and Integration Plan

- Due Week 14
- This document will contain information about the plan for ensuring that the Amplify application is well-tested so that errors can arise and be fixed before the final delivery of the project. It will detail the plan for unit testing as well as the description of the unit tests (listing all of the precise tests run). It will also outline the plan and description of the integration tests, which will test the application at a higher level than

the unit tests, which test specific functions or elements of the application.

4.1.1.13 User's Manual Final Updates

- Due Week 15
- This deliverable will contain instructions as to how an outside user will be able to use our application. It will contain thorough documentation and is intended to guide the user through the installation, use, uninstallation, features, and functionality of Amplify. It will tell the user how to use the application and what to do in the event of something unexpected happening.

4.1.1.14 Oral Status Report

- Week 15
- Oral status reports will give the team an opportunity to go over what has been done thus far and what still needs to be done, either in the current sprint or farther along in the schedule. It will also allow the team to update the client (instructor) on the current progress of the project.

4.1.1.15 FINAL Project Presentation

- Due Week 15
- This presentation will act as an update to the class and instructor about the finalized version of Amplify at its formal delivery. It will contain details about the full functionality of the application, challenges the team faced in producing the application, and a demonstration of the use of the application.

4.1.1.16 FINAL Product Delivery (Final Report and Code)

- Due Week 16
- This will contain the delivery of the entire finalized project, including its code, documentation, and a final report on the status of the final applications functionality.

4.2 Project Resources

4.2.1 Hardware Resources

4.2.1.1 Amplify development will require a computer that is able to support an iPhone or Android emulator, or an iPhone or Android device to test on.

4.2.1.2 Amplify development will require a computer that is able to run a Node.js server, and support React Native development, preferably through an IDE.

4.2.1.3 Amplify development will require a computer that is able to run TypeScript.

4.2.1.4 Amplify development will require a computer that is able to run a Postgres DB instance for a local database for development purposes.

4.2.2 Software Resources

4.2.1.1 Amplify development will require an iPhone or Android emulator, or an iPhone or Android device to test on.

4.2.1.2 Amplify development will require a Node.js server, and React Native development, preferably through an IDE.

4.2.1.3 Amplify development will require TypeScript.

4.2.1.4 Amplify development will require Postgres DB instance for a local database for development purposes.

4.2.2.5 Amplify development will require a text editor or IDE for coding and editing purposes.

4.2.2.6 Amplify development will require git and GitHub access.

4.3 Project Organization

4.3.1 Ian Lizarda: Frontend Team Lead

Ian is responsible for all project management and assignment of tasks for the Frontend. Ian also manages pull requests.

4.3.2 Donovan Moini: Backend Team Lead

Donovan is responsible for all project management and assignment of tasks for the Backend. Donovan also manages pull requests.

4.3.3 Serena Zafiris: Frontend Developer and Lead UX Designer

Serena assists with the development of the Frontend. She also performs UX research and creates and manages preliminary designs.

4.3.4 Alexia Filler: Backend Developer

Alexia assists with the development of the Backend.

4.3.5 Ben Kern: Backend Developer

Ben assists with the development of the Backend.

4.3.6 Eddie Azinge: Team Technical Advisor

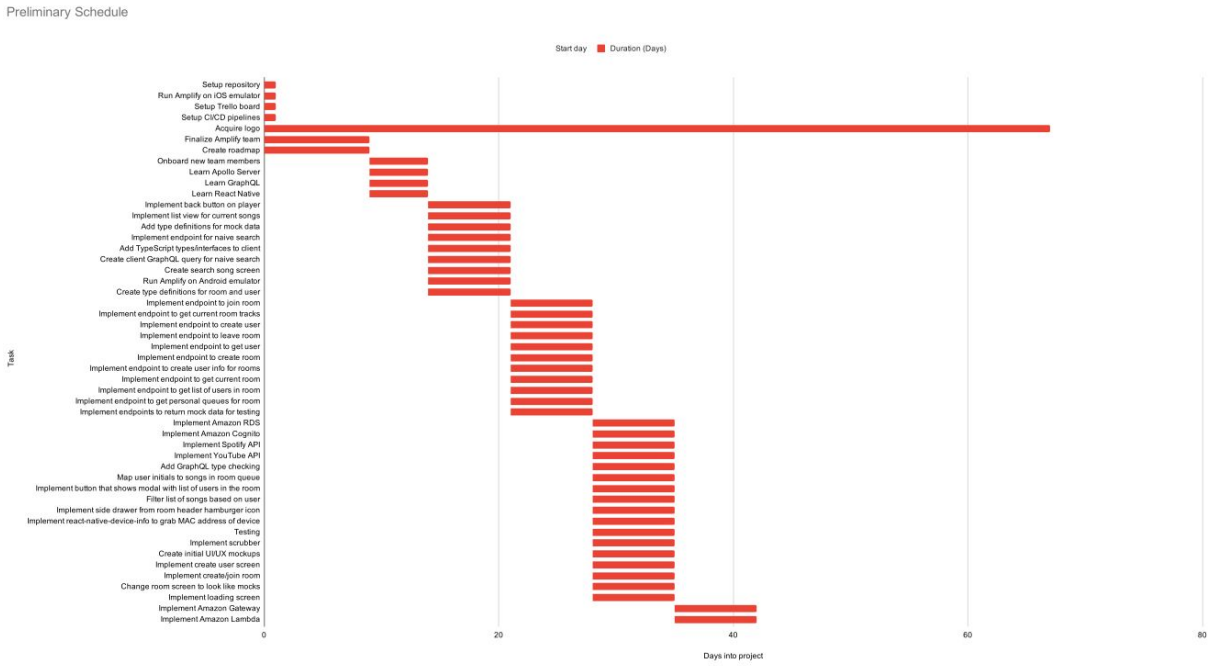
Eddie advises and relays information to Donovan and Ian. He provides technical knowledge and created the initial infrastructure setup.

4.3.7 Masao Kitamura: Contractor

Masao performs supplemental work on the frontend for non-vital assets that are in the backlog.

4.4 Project Schedule

4.4.1 PERT / GANTT Chart



4.4.2 Task / Resource Table