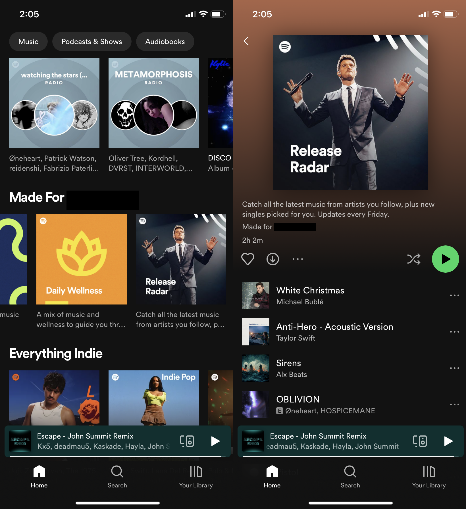
COMP 1004 – Computing practice

2023/2024

Music player application (Theme: ENTERTAINMENT)

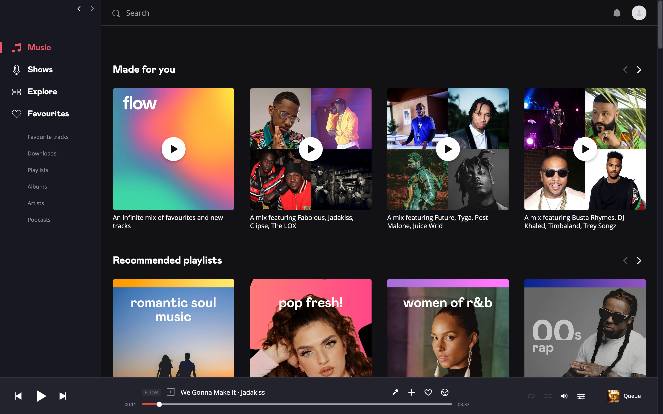
Introduction

In this current day most people listen to music either online or through downloading and storing it on their devices. Most music application software allows for users to create playlists, favourite music and see recommended songs based on what users are listening too. I will be developing an online music player that if possible, will allow users to search for music online, upload their own music, create playlists and download songs for offline listening. The report I create will show the stages of the software development lifecycle along with the software requirements, including the user stories and the architecture of the system. The report will then describe my sprint plans so that I know what to do when planning my implementation. I will also be evaluating the parts of my project that I add to my report. This will allow me to filter out bits that may not be relevant to my project. This report will contain labelled graphs and diagrams to make it easier for readers to understand what my web application will involve and how it will work.

The image shows an already existing music application, known as Spotify. It allows users to stream music as well as download and upload their own songs.

Spotify uses a unique colour scheme for its branding and uses an easy on the eyes menu to allow for people of all ages to have good accessibility.

Source: Google images

Another example is Deezer, which is a similar music software that uses online music streaming and other user uploads to allow people to listen to music. It gives recommendations based off what people are listening too and I will be taking inspiration from these sorts of features. Just like Spotify it allows users to create playlists and filter through music using categories such as artists etc.

The menu also consists of bright eye-catching colours against a darker background.

**Software development lifecycle (SDLC)**

The software development lifecycle allows for a methodical and easy approach to developing any computer programs. There are 5 stages to this process:

* Requirements
* Design
* Implementation
* Testing
* Maintenance

There are also many types of SDLC, each one suited to a certain type of project better. They vary in how the order of the instructions are executed and the amount of time spent on each stage.

Requirements Analysis

This is the first stage of the software development lifecycle, it is about gathering all the requirements needed to start development on your project. These can be split into 3 parts which include, Functional, Non-Functional and usability. Functional requirements are requirements that include the tasks that the product performs and how it reacts to inputs. Non-functional requirements describe the restrictions and restraints of a piece of software and how well it can meet its functional requirements. The usability requirements are the last set that are needed to describe how easily a piece of software can be used. There is a fourth set called external requirements, these are a last thought and include the social, ethical and legal considerations of the project.

Design

After the requirements have been fulfilled you move onto the design stage which involves two parts. These are called detailed and style both being used for different parts of the project’s development. Style is used to design the architecture and the basic layout of how the software will run and function. An object-oriented paradigm is a way of viewing the software during development and uses classes with set attributes and methods. This makes code reusable.

During the design phase you also need to create diagrams in the UML such as class diagrams, state diagrams and sequence diagrams. Class diagrams allow users to present classes and objects in software.

Implementation

The implementation stage is where you take what you developed in the design section and start putting it into code. This is usually the most important part as you are now actually writing the program. The code language for the webpage will be using HTML. During this stage it will be useful to refer back to you design ideas and class graphs.

Testing

The testing phase happens once the programmer is happy with the implementation and programming stage. The programmer then runs multiple tests on the software to weed out any bugs and errors that will need to be fixed. The testing phase can sometimes happen alongside the implementation phase as the programmer will discover bugs during development. Some errors can be worse than others and depending on the severity it might make the programmer have to revert back to the implementation or design stage to recode or rethink that entire part.

Maintenance

The final part of the lifecycle is known as the maintenance stage and is used post program launch. This stage is all about maintaining the software as the system around it updates. For example, windows operating system updating to a later version my make the current version of the software incompatible meaning it will need an update to fit with the new system. This stage might not always be about just keeping an application up to date, there will be times where the software may need some minor bug fixes are updates/patches to fix errors that will pop up throughout its lifetime.

Maintenance may be the smallest step in the SLDC when it comes to adding content but it will be the longest lasting one as it will be in effect throughout a program’s lifetime. You see examples of it everyday when the majority of apps on our phones see updates regular. It’s the same for computers, where the operating system is constantly seeing patches and small updates.

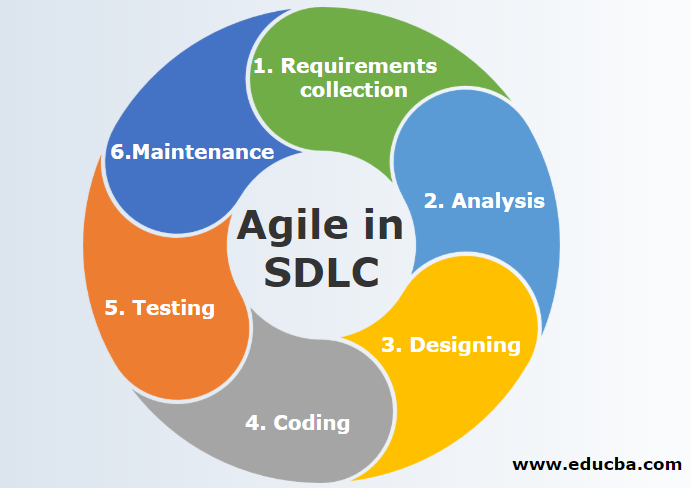
Types

When it comes to types of the SDLC there are many to choose from. The list includes names such as the Waterfall mode, Increment model, Spiral model and the V-model. These all vary in how their stages are laid out and what order they are executed. Some of them allowing for very flexible stages where as others are more fixed in approach and are a lot stricter and more unforgiving when mistakes are made in the cycle.

For my project we have been assigned to use the AGILE software development life cycle model.

Agile

The Agile SDLC is a very requirement focussed one that relies on users seeing quick results coming from each stage. The agile approach uses periods called sprints that are normally done in two-week time slots. In this time a small amount of the project is completed, this can range from any stages of the SDLC. Then after that period, there is usually a meeting where many developers come together in an “Agile team”. They discuss progress and points such as what did the developers change or what they added to there base project idea. User stories and a backlog are a stage only present in this type of SDLC, which is seen featured later on in my report. User stories are used to describe what features a piece of software will contain in an almost first-person user perspective. These user stories may not always be fulfilled but they give a rough idea to the user of what the program should feature. The project backlog is the other unique planning stage of Agile that is a document which contains all the current features of a sprint that need to get worked on. All the features being worked on and all the completed features.



This image shows a basic graph that lays out the Agile SDLC. Agile is also a method that allows user to go back and change things in each step at any point in time. Unlike some other types that don’t allow users to go back and make changes to earlier parts. Once you have done it, you must go through with it.

**Design Documentation**

Project description

In a modern age most people use mobile devices or computers for entertainment. Listening to music is one of the larger ways people like to spend their free time, whether it’s working, studying or just relaxing. Music in the background always helps. Everyone should have access to this sort of thing.

As stated in my introduction my project will be a single webpage music application. Where users can search for different songs and listen to them online. The file inputs and outputs will come from the user being able to input song names/song links and the title, artist and release date etc being outputted onto a file. The software will come with many features such as being able to create playlists out of multiple songs that a user may like. The application will have an introduction screen login that will allow a user to have an account where they can then access all their recent songs, playlists etc. This will also allow for multiple people to use the webpage application on the same device. The login page will allow for users to ask for a password reset etc.

To create this, I will be using an API and embeds maybe taken from existing software, such as a music database from Spotify or amazon music.

User stories

* As a User I should be able to reset my password if I forget or lose it.
* As a User I should be able to easily open and close the webpage.
* As a User I want to be able to log on to the music software, allowing me to see my saved songs/playlists. My audio settings and uploads should be saved to my account.
* As a User I want to be able to search for any released song and play it or download it for offline use.
* As a User I want to be able to create playlists with user inputted titles and descriptions.
* As a User I want to be able to upload my own music, either publicly or have it saved as private for my own use.
* As a User I want to be able to save songs as favourites and “listen for later”. These categories should be easily accessible and distinguishable from each other.
* As a User I should be able to sort songs by certain criteria such as date released, artist name and alphabetical order.
* As a User I should be able to input a song link and in return receive a file that outlines the songs data, such as title, album, artist and release date.

Backlog

The application backlog is part of my sprint planning phase. This will be used to store all my user stories throughout the products development. Including tasks that need to be completed to represent my user stories. There will be lower priority tasks and higher priority ones, this intern helping decide what I need to work on next. The will be a section for “NEED TO DO”, “DONE” and “CURRENTLY WORKING ON”. The backlog should be constantly updated. My backlog is going to be created in Microsoft planner.