Project Overview

In this project, you will create an application that will manage playlists for a Home Audio System that streams songs to various locations in a home. In teams of four or five students, you will gather requirements, architect and design a software solution to satisfy those requirements, implement the system, thoroughly validate that the system is satisfying the requirements, and develop a release pipeline to automate the software delivery process.

You will deliver the system in iterations throughout the term. Several deliverables are to be submitted at checkpoints throughout the term as described below.

1 User Story for Home Audio System

A modern home very often has audio entertainment systems in more than one room in the home, e.g., in the living room, in the basement, in the kitchen, and in the bedrooms. The Home Audio System (HAS) manages what music is played in which room of the home from a central location. The HAS maintains a library of songs grouped by album or artist and keeps track of song title and duration, a song's position on an album, the genre of the album as well as release date, and an artist's name. The user of HAS sets up the locations with audio systems in the home. HAS also offers the user the ability to create playlists, which contain a list of songs in a specific order. When the user wants to play music, the user assigns a playlist, album, or song to a location in the home. It is also possible to stream the same music to several locations at the same time. For each location, the volume can be controlled individually and muted temporarily.

2 Scope and Technology Constraints

Your application must support the scenarios described in the user story except that the actual streaming of songs to the various locations in a home is out of scope for this project.

Interoperability is a very important characteristic for this system. Your application must support laptop/desktop machines, mobile devices, and web browsers. Your laptop/desktop application must be implemented in Java with the Java Swing framework for the user interface. Your mobile application must be implemented in Java on the Android platform using the UI development framework that comes with Android Studio. Your web application must be primarily written in Php but other web technologies are also welcome (e.g., CSS or Javascript for the user interface).

The data of your application must be saved in text files (e.g., XML) or in a database. Note that the three applications (laptop/desktop, mobile, web) are not required to interact with each other, e.g., any changes to the application data made on your mobile phone do not need to show up in your web application.

3 Deliverables

This section gives an overview of the deliverables including their due dates. More details for Deliverable 1 are available in the following section. The details of Deliverable 2/3/4/5/6 will be posted at the latest on the due date of Deliverable 1/2/3/4/5, respectively.

Deliverable 1 – Requirements Document and Prototype (10%) (due February 22, 2016 23:30)

- Functional and non-functional system requirements

- Domain model
- Use cases
- Requirements-level sequence diagram for “Add Album” use case
- Source code of prototype implementation of “Add Album” use case on each supported platform
- Implementation-level sequence diagram for “Add Album” use case for each supported platform
- Work plan for remaining iterations

Deliverable 2 – Design Specification (5%) (due March 07, 2016 23:30)

- Description of architecture of proposed solution including block diagram
- Description of detailed design of proposed solution including class diagram
- Update of work plan

Deliverable 3 – Quality Assurance Plan (5%) (due March 21, 2016 23:30)

- Description of unit testing
- Description of component testing
- Description of system testing
- Description of performance/stress testing
- Update of work plan

Deliverable 4 – Release Pipeline Plan (5%) (due March 28, 2016 23:30)

- Description of release pipeline
- Update of work plan

Deliverable 5 – Presentation (8%) (due April 11/12/14, 2016)

- Each group will give a presentation about their project to the class
- A random draw will be held in class on February 23 to determine the order of presentations

Deliverable 6 – Final Application (12%) (due April 15, 2016 23:30)

- Source code of full implementation on each supported platform

Deliverable 1 – Requirements Document and Prototype

The report for this deliverable contains the requirements document, three implementation-level sequence diagrams, and the work plan as described below. You are also required to submit the zip files of your three prototype applications.

1 Requirements Document

The requirements document contains a list of functional and non-functional system requirements. Each requirement must have an ID. In addition, the priority of each requirement must be indicated. The requirements document also contains a domain model defined with Umple and a use case diagram. Each use case must be described in natural language, including the name of the use case, the main flow of the use case, and any alternative/exceptional flows as needed. The actor(s) involved in a use case must also be stated. Furthermore, each use case must indicate which requirements are covered by the use case. Finally, a requirements-level sequence diagram for the "Add Album" use case must be included in the requirements document.

2 Prototype

The prototype to be submitted covers the "Add Album" use case on each supported platform. You must use Umple to generate Java and Php code from the domain model, and use the generated code in each of your applications. In addition, you are required to submit the implementation-level sequence diagram for the "Add Album" use case for each of the three supported platforms.

3 Work Plan

Your work plan must define the remaining iterations of your implementation until the final deliverable at the end of the term. For each iteration, indicate which requirements are addressed, provide an estimate for the effort needed to complete the iteration, and the date when the iteration is to be completed.

Submission

The project is to be done in teams of FOUR or FIVE students. Your team will be assigned a GitHub repository, which you will use for your work on this project. The instructor and the TAs also have access to your GitHub repository. You are required to sign up to one of the project groups in myCourses by ***Thursday, February 11, 2016 at 23:30***. In addition, you are required to send the instructor an email with the GitHub user names for all your team members. This also needs to be done by ***Thursday, February 11, 2016 at 23:30*** at the latest, but you are strongly advised to do this as soon as possible to get access to your GitHub repository as early as possible. Note that you will lose marks if you miss this deadline (see Marking Scheme below).

For Deliverable 1, your team is required to hand in the ***report*** and ***three individual zip files*** by ***Monday, February 22, 2016 23:30***. The report must be either an MSWord file or a PDF file. If you are using an application other than MSWord, convert your report first to either a PDF file or a DOC(X) file. For the zip files of your desktop and web applications, use the *Export* feature of Eclipse to create an *Archive File* (Export – General – Archive File) of the project. For the zip file of your mobile application, simply zip up the folder of your Android project with your favorite archiving tool. If you realize that you need to make changes to the submission of one of your applications, do not resubmit only the file(s) that have changed, but rather resubmit another complete zip file.

Each team member must make contributions to the deliverable. A team member who does not contribute to the deliverable receives a mark of 0 for the deliverable. A team member may optionally email a confidential statement of work to the instructor before the due date of the deliverable. A statement of work first lists in point form the parts of the deliverable to which the team member contributed. In addition, the statement of work also describes whether the work load was distributed fairly evenly among the team members. A statement of work may be used to adjust the mark of a team member who is not contributing sufficiently to the deliverable. It is not necessary to send a statement of work, if a team distributed the work for the deliverable fairly evenly and each team member contributed sufficiently.

Marking Scheme

Part of Assignment	***Marks***
On-time submission of GitHub account names	5
Functional and non-functional system requirements	5
Domain model	10
Use case diagram	2
Use cases	8
Requirements-level sequence diagram for “Add Album” use case	5
Desktop/laptop application:	20
Implementation of functionality for “Add Album” use case	10/20
Implementation of persistence for “Add Album” use case	4/20
Implementation of Validation Checks for “Add Album” use case	4/20
Implementation-level sequence diagram for “Add Album” use case	2/20
Mobile application:	20
Implementation of functionality for “Add Album” use case	10/20
Implementation of persistence for “Add Album” use case	4/20
Implementation of Validation Checks for “Add Album” use case	4/20
Implementation-level sequence diagram for “Add Album” use case	2/20
Web application:	20
Implementation of functionality for “Add Album” use case	10/20
Implementation of persistence for “Add Album” use case	4/20
Implementation of Validation Checks for “Add Album” use case	4/20
Implementation-level sequence diagram for “Add Album” use case	2/20
Work plan for remaining iterations	5
Total Marks:	100
The total mark may be adjusted based on the actual contributions of a team member to the deliverable.	