Personal Archival Companion (P.A.C)

Project Team: Cody Cothern, Quincy Hinson, Jacob Oleson, Andrew Menezes

Table of Contents

**1. Project Definition (**100 - 200 words**)** – *Group responsibility*

* Why (it is needed)
* What (is the goal of the project)
* How (how will it be achieved)

**2. Project Requirements** – *Group responsibility*

* Functional
* Usability
  + User interface
  + Performance
* System
  + Hardware
  + Software
  + Database
* Security

**3. Project Specification** – *Group responsibility*

* Focus / Domain / Area
* Libraries / Frameworks / Development Environment
* Platform (Mobile, Desktop, Gaming, Etc)
* Genre (Game, Application, etc)

**4. System – Design Perspective** – *Group responsibility*

* Identify subsystems – design point of view
  + Illustrate with class, use-case, UML, sequence ..... diagrams
  + Design choices (Optional)
* Sub-System Communication (Diagram and Description)
  + Controls
  + I/O
  + DataFlow
* Entity Relationship Model (E-R Model)
  + Example - <https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model>
* Overall operation - System Model
  + Simplified Sub-system to System interaction

**5. System – Analysis Perspective** – *Group responsibility*

* Identify subsystems – analysis point of view
* System (Tables and Description)
  + Data analysis
    - Data dictionary (Table - Name, Data Type, Description)
  + Process models
* Algorithm Analysis
  + Big - O analysis of overall System and Sub-Systems

**6. Project Scrum Report -** *Group Responsibility*

* Product Backlog (Table / Diagram)
* Sprint Backlog (Table / Diagram)
* Burndown Chart

**7. Subsystems**

**7.1 Subsystem 1** – Name 1 - *Individual responsibility*

* Initial design and model
  + Illustrate with class, use-case, UML, sequence ..... diagrams
  + Design choices
* Data dictionary
* If refined (changed over the course of project)
  + Reason for refinement (Pro versus Con)
  + Changes from initial model
  + Refined model analysis
  + Refined design (Diagram and Description)
* Scrum Backlog (Product and Sprint - Link to Section 6)
* Coding
  + Approach (Functional, OOP)
  + Language
* User training
  + Training / User manual (needed for final report)
* Testing

**7.2 Subsystem 2** – Name 2 - *Individual responsibility*

* Initial design and model
  + Illustrate with class, use-case, UML, sequence ..... diagrams
  + Design choices
* Data dictionary
* If refined (changed over the course of project)
  + Reason for refinement (Pro versus Con)
  + Changes from initial model
  + Refined model analysis
  + Refined design (Diagram and Description)
* Scrum Backlog (Product and Sprint - Link to Section 6)
* Coding
  + Approach (Functional, OOP)
  + Language
* User training
  + Training / User manual (needed for final report)
* Testing

**7.3 Subsystem 3** – Name 3 - *Individual responsibility*

* Initial design and model
  + Illustrate with class, use-case, UML, sequence ..... diagrams
  + Design choices
* Data dictionary
* If refined (changed over the course of project)
  + Reason for refinement (Pro versus Con)
  + Changes from initial model
  + Refined model analysis
  + Refined design (Diagram and Description)
* Scrum Backlog (Product and Sprint - Link to Section 6)
* Coding
  + Approach (Functional, OOP)
  + Language
* User training
  + Training / User manual (needed for final report)
* Testing

**7.4 Subsystem 4** – Name 4 - *Individual responsibility*

* Initial design and model
  + Illustrate with class, use-case, UML, sequence ..... diagrams
  + Design choices
* Data dictionary
* If refined (changed over the course of project)
  + Reason for refinement (Pro versus Con)
  + Changes from initial model
  + Refined model analysis
  + Refined design (Diagram and Description)
* Scrum Backlog (Product and Sprint - Link to Section 6)
* Coding
  + Approach (Functional, OOP)
  + Language
* User training
  + Training / User manual (needed for final report)
* Testing

**8. Complete System** – *Group responsibility*

* Final software/hardware product
* Source code and user manual – screenshots as needed - Technical report
  + Github Link
* Evaluation by client and instructor
* Team Member Descriptions

1. Project Definition

People are always on the go; they go to and come from work, travel for business and pleasure. To entertain themselves during these typically boring periods of travel, people tend to read, either physical books or ebooks, or, if their hands are busy, listen to audiobooks or podcasts. A problem arises when one has to many books and files to easily keep track of.

Now it is true that many applications exist to help make handling and organising all of this entertainment easier, but most tend to focus on only one type. Most of these tend to be clunky, slow, not user-friendly, or lack ease of life features. No one wants to have to worry about having dozens of applications to handle a few tasks, especially if those applications are not easy and intuitive to use. Our goal with this project is to take all of these problems, bind them together, and solve them with one easy-to-use application. The method to solving this will be two-fold.

First, we will create a desktop application that acts as the repository and hub for all of the user’s ebook, audiobooks, and podcast files, as well as keeping track of the user’s physical books, both owned and desired. Secondly, we will create a mobile application that will sync with the desktop version and allow the user to access their aforementioned files wherever they happen to be. The mobile app will also include a way to scan the barcode of physical books to automatically look up the books for addition to the user’s personal archive.

1. Project Requirements
   1. Functionality
      1. Catalog of user’s ebooks, audiobooks and podcasts.
         1. User can manually enter a book’s information to catalog it
         2. User can scan a physical book’s barcode with their mobile phone to add that book to their collection
         3. User can search the internet for a book to catalog it.
      2. User can add/remove books from their wishlist
      3. File transfer between desktop version and mobile version.
         1. User will be able to specify which ebook/podcasts will be added to their mobile phone
      4. Application will use the newest information from either the desktop app or mobile app to ensure that the information is up to date across devices
   2. Usability
      1. User Interface
         1. Simple interface that is easy to understand
         2. Files containing users’ contents will be appropriately labeled and accessible
         3. Basic settings will be customizable and accessible to the user.
      2. Performance
         1. Application will be responsive and utilize a SQL Database to load and manage their personal libraries
         2. In the event of errors or shut downs the user will be notified with an adequate explanation of why the error occurred
   3. System
      1. Hardware
         1. A PC running the Windows OS
         2. A mobile android device
      2. Software
         1. The Java runtime environment for Windows
   4. Security
      1. Prevent against SQL Injections to protect users’ accounts and their library of media
2. Project Specification
   1. Focus - Users with a large collection of physical books, ebooks, audiobooks, or podcasts
   2. Platform
      1. Windows PC
      2. Android mobile