React Web App – Decisioning Dashboard

Benjamin Roberts

Level 4 Software Development

Contents

[Table of Figures 3](#_Toc170156481)

[Introduction & Requirements 4](#_Toc170156482)

[Stakeholders Involved 4](#_Toc170156483)

[Design 5](#_Toc170156484)

[Development 6](#_Toc170156485)

[CRUD - Database Connection 6](#_Toc170156486)

[Personal Data (PII) Protection 7](#_Toc170156487)

[Credentials within the Web Application 8](#_Toc170156488)

[Testing the program 8](#_Toc170156489)

[Unit Tests 8](#_Toc170156490)

[Debugging Errors 8](#_Toc170156491)

[Review with Employer 8](#_Toc170156492)

[Conclusions 8](#_Toc170156493)

[Project Outcome 8](#_Toc170156494)

[Future Additions 8](#_Toc170156495)

[References 8](#_Toc170156496)

# Table of Figures

**No table of figures entries found.**

# Glossary of Acronyms

Below is a series of Acronyms and terminology definitions that will be useful for understanding concepts throughout the project

## Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Name** |
| MI | Management Information |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Terminology

|  |  |
| --- | --- |
| **Term** | **Description** |
| Raw Data |  |
| Scorecard |  |
| Summary Data |  |
|  |  |
|  |  |
|  |  |

# Introduction & Requirements

With my team being responsible for developing and maintaining our Credit Decisioning system, we require various kinds of Management Information (MI) and views of the data used by the platform (both in the aggregate and within individual applications). This is also important for monitoring the performance of the platform.

Today, we use various tools to monitor and report on the performance of the system e.g.:

* Ad-Hoc analysis done via querying a SQL Database
* Summarised Microsoft Excel Reports
* Microsoft Power BI reports
* Manually extracting and reviewing raw data (typically via SQL)

All these tools are useful for their specific purposes but what they don’t do is give us a presentable view of the low-level data used in making a decision on an application. We could license other software for this purpose e.g. the Credit Bureau providers Experian, TransUnion and Equifax all provide software to present the data they hold on an applicant. These however require software licenses to be purchased. I believe we could use the React Framework to build a similar tool that would be more bespoke to our business needs.

## Project Description

Using the React Javascript Framework, design and develop a simple Web Application to:

1. Leverage the Java Decisioning Application I developed in a previous project & it’s logged output (Link: [GitHub - Java Credit Decisioning Program](https://github.com/btr6566/qam1_java_decisioning_app))
2. Display a list of recent applications made to the Credit Decisioning system
3. Allow a user to click on a specific application to view the data used for it in more detail
4. Provide reference material for the data definitions of the data used (i.e. a Data Dictionary function)
5. Provide some high level dashboards on applications volumes

## Acceptance Criteria

1. Wireframe of the design of the website is provided
2. Final product aligns to the wireframe
3. Intuitive user interface created & evidenced via feedback from technical & non-technical stakeholders
4. The web app is able to query a database where logs from the Decisioning program/software are retained
5. Low level detail of the data used in the application, including:
   1. Data used for running Scorecards + their results
   2. Summary data on an applicants Credit file
   3. Raw data used to create the above summary data

# Summary of Stakeholders Involved

Below is a summary of the Stakeholders I engaged for support and feedback as I developed the project, along with the communication methods involved:

* **F** = Face to Face
* **IM** = Instant Messaging via Microsoft Teams
* **C** = Call via Microsoft Teams
* **E** = Email

|  |  |  |
| --- | --- | --- |
| **Stakeholder** | **Methods** | **Input** |
| Credit Systems | **F, IM, C** | * Sign-off for project with Line Manager |
| Credit Risk Strategy |  | * Details on what views exist in other tools today |
|  | **E** |  |
|  | **IM** |  |

# Design

To give myself a starting point, I referred to the Software Development Life Cycle, which as “design” as a key starting point for build a piece of software (Amazon AWS, 2024).

I needed a design to work towards before starting the coding of my web app. To do this, I started with a draft wireframe and then expanded upon it with a 2nd iteration.

## Initial Draft (Home Page only)

To keep a simple process to start, I drew a basic template on a white board for that I had in mind for a “home” page:

A white board with a drawing on it

Description automatically generated

Figure 1- Initial draft Wireframe drawn on Whiteboard

Doing this simple approach gave me a way to get a starting point for what I wanted the layout of my application to look like, without having to decide on specific details (like styling).

## Wireframe

# Implementation / Development

## Setting up Dependencies

## CRUD - Database Connection

### Personal Data (PII) Protection

~~This access management configuration can also be used for another purpose: protecting Personal Data (PII). Credit Decisioning Systems like this must collect personal data to be able to conduct a credit search on a person e.g. Experian’s DelphiSelect API requires at least:~~

* ~~Full name of the person in question~~
* ~~Date of Birth~~
* ~~At least the current address of the person in question~~
* ~~Previous addresses are also often needed if the person in question has resided at their current address for less than 3 years~~

~~(Experian, 2024)~~

~~Each of these Data Points are protected under the UK's Data Protection Act 2018, which also codifies the GDPR into UK Law (Data Protection Act, 2018). The program is currently collecting the Full Name of the user as a starting point to allow a full API integration in future. This means the program must be mindful of Data Protection/GDPR requirements.~~

~~Using a SQL databases within the program creates the risk of SQL Injection, where malicious input into the program can lead to arbitrary execution e.g. a user passes in a SQL command instead of a name (W3Schools, 2024). This is where the SELECT and INSERT permissions assigned above come into effect. The Full Name collected in the application is only saved to the “dbo.JavaDecisioningHistory” , which the integration user only has INSERT permissions on. This prevents a malicious user from entering a query to retrieve data from this table, as the database permissions will result in the query being rejected.~~

~~The only other permission the Integration User has been granted is the SELECT permission on the “Delphi.VW\_DelphiPremiumValueData” and the “Delphi.VW\_DelphiSummaryData” views. These are the only other views required by the program, so by limiting the access like this the possibility of risk incurred by SQL Injection is minimised and therefore acts as a control to protect Personal Data.~~

### Credentials within the Web Application

# Testing the program

## Unit Tests

## Debugging Errors

# Review with Employer

# Conclusions

## Project Outcome

## Future Additions

# References

Amazon AWS. (2024). What is SDLC (Software Development Lifecycle)?. [Online]. AWS. Last Updated: 2024. Available at: <https://aws.amazon.com/what-is/sdlc/#:~:text=The%20software%20development%20lifecycle%20(SDLC,expectations%20during%20production%20and%20beyond>. [Accessed 28 June 2024].