# 

**Individual Project – Payroll system**

**with Oracle Database**

Kyle Simon Chaplin

Text

Description automatically generated

# Abstract

The Payroll System with Oracle Database project presents a comprehensive solution for efficient organisational payroll management. This report details the design and implementation of a system that automates the calculation of pay, taxes, and benefits for employees and seamlessly delivers personalised payslips via email on payday. The project leverages the power of Oracle Apex to provide a robust, secure, and user-friendly application.

Content

[Abstract 3](#_Toc164895922)

[Introduction 6](#_Toc164895923)

[Objectives 6](#_Toc164895924)

[Aims 7](#_Toc164895925)

[Project Management 7](#_Toc164895926)

[Development 8](#_Toc164895927)

[Planning 8](#_Toc164895928)

[Milestone 1 9](#_Toc164895929)

[Milestone 2 9](#_Toc164895930)

[Main Submission 9](#_Toc164895931)

[Research 10](#_Toc164895932)

[Low Code Platforms 12](#_Toc164895933)

[Mendix 12](#_Toc164895934)

[Microsoft Power Apps 12](#_Toc164895935)

[Oracle Apex 13](#_Toc164895936)

[OutSystems 14](#_Toc164895937)

[Competing Applications 14](#_Toc164895938)

[BambooHR 15](#_Toc164895939)

[Sage Payroll 15](#_Toc164895940)

[Wave 16](#_Toc164895941)

[Data Protection and Privacy 16](#_Toc164895942)

[System Design and Implementation 19](#_Toc164895943)

[Prototyping 19](#_Toc164895944)

[Dashboard 19](#_Toc164895945)

[Schedule 19](#_Toc164895946)

[Payroll 20](#_Toc164895947)

[People 21](#_Toc164895948)

[Profile 21](#_Toc164895949)

[Help 22](#_Toc164895950)

[Table Design 23](#_Toc164895951)

[Entity Relationship Diagram (ERD) 23](#_Toc164895952)

[Oracle APEX Prototype 25](#_Toc164895953)

[Application Development 27](#_Toc164895954)

[Advantages of Java and JavaFX 27](#_Toc164895955)

[Technical Explanation 28](#_Toc164895956)

[Libraries and Tools 28](#_Toc164895957)

[Database setup 29](#_Toc164895958)

[Implementation 30](#_Toc164895959)

[Code improvements 50](#_Toc164895960)

[Testing plan and results 52](#_Toc164895961)

[Testing table 52](#_Toc164895962)

[System usability 56](#_Toc164895963)

[Reflection 61](#_Toc164895964)

[Aims 61](#_Toc164895965)

[Planning 62](#_Toc164895966)

[Future improvements 63](#_Toc164895967)

[Conclusion 64](#_Toc164895968)

[References 65](#_Toc164895969)

[Appendix 68](#_Toc164895970)

[Risk Mitigation 68](#_Toc164895971)

[Heuristic Evaluation Workbook - 1 72](#_Toc164895972)

[Visibility of System Status 72](#_Toc164895973)

[User Control and Freedom 73](#_Toc164895974)

[Consistency and Standards 73](#_Toc164895975)

[Heuristic Evaluation Workbook - 2 77](#_Toc164895976)

[Visibility of System Status 77](#_Toc164895977)

[User Control and Freedom 78](#_Toc164895978)

[Consistency and Standards 78](#_Toc164895979)

[The System Usability Scale (SUS) 82](#_Toc164895980)

[Ethics Form 83](#_Toc164895981)

[Changes from milestone 1 88](#_Toc164895982)

# Introduction

In today’s dynamic and digitally driven business environment, efficient systems, including payroll management, are paramount to an organisation’s success. The accurate calculation of employees’ salaries, taxes, and benefits not only ensures the financial well-being of the workforce but also stands as a testament to an organisation’s commitment to its employees.

The development of the Payroll System aims to address the growing complexities of organisations and provide an integrated, automated, and user-friendly solution. Oracle Apex will be used to create a prototype of the system, and Java is the chosen language to create a versatile and scalable application that streamlines the payroll process, ensuring it is accurate, efficient, and secure. This application not only calculates the monthly pay, taxes, pensions, and benefits but also generates personalised payslips that are delivered directly to the employee on payday.

This report offers a comprehensive insight into the design, development, and implementation of the Payroll System. Each section explores the various functions of the application, from managing employee information to automatic tax calculations and the seamless email notification system. Furthermore, it discusses the testing and quality assurance procedures to ensure a reliable and efficient product.

# Objectives

#### Research existing payroll systems and possible tools to create my application

There are many existing payroll systems, and both open and closed source applications will be examined to gain a better understanding of required features and what can be improved.

#### Design and develop the payroll system

Using the knowledge gained from research to design and develop a Payroll System that can efficiently manage the payroll process.

#### Testing and Quality Assurance

Conducting rigorous testing will ensure my application is of high quality, helping identify and rectify flaws or vulnerabilities.

# Aims

#### To develop a payroll system that enhances the employee experience

The application aims to enhance the overall employee experience by providing personalised payslips and a convenient and easy way to access payroll information. This will also reduce the time and effort required for calculations and record-keeping, improving efficiency.

#### Always think of data protection and privacy

Recognising and adhering to strict data protection and privacy regulations such as GDPR. Data protection and privacy is a right therefore ensuring data in the application processes is appropriately secured is highly important.

#### To develop with future improvements in mind

Aimed to design a scalable application capable of accommodating the evolving needs of organisations as they grow and change over time.

# Project Management

Effective project management is essential for the successful execution of this project. A well-structured plan and meticulous organisation are crucial to ensuring that the project achieves its objectives within the specified time frame. To reduce the possibility and effect of scope creep I have outlined the scope and deliverables of this project below:

* **Conduct research on existing payroll systems:** Explore existing payroll systems and technologies to gather insight for potential solutions in my system.
* **Define core system processes:** Identify and outline the fundamental processes that the system will manage, ensuring they align with the project goals.
* **Design and create a prototype**:Develop a representation of the system concept to gain early insights into functionality and feasibility, adjusting as needed.
* **Design a database structure:** Develop a database structure to store and manage system data effectively.
* **Adhere to UK data protection and privacy legislation:** Ensure the system meets the standards of data protection and privacy legislation in the UK.
* **Produce a comprehensive report:** Create a detailed report outlining the entire project journey, including planning, designing, creating and testing the application.

## Development

There are several options when considering a development cycle. For this project, a Hybrid Method, focusing elements of the Waterfall and Agile methods, offers a planned yet adaptable approach. This Hybrid approach allows leverage over the structured planning and documentation aspects of the Waterfall model whilst also incorporating the adaptability of agile development.

This approach will entail setting clear deadlines for specific features and tasks. However, understanding the need for adaptability throughout development, testing will be completed and adapting the plan and/or application may be necessary. I will complete tasks in a linear approach, understanding that refinement can be completed at a later date, such as the user interface. This approach will ensure the highest quality end product in the time specified.

## Planning

A Gantt chart has been used to plan and visualise the project’s schedule and tasks. The Gantt chart below displays the timeline for this project, broken down into key milestones:

|  |
| --- |
| Figure 1 – Gantt chart |
|  |

### Milestone 1

By the 10th of November 2023, a layout for my report has been completed, identifying the aims and objectives of the project, research and documentation in the form of a literature review, creating a simple prototype of the application and completing an ethics form.

### Milestone 2

The second milestone is to be completed by the 9th of February 2024. It includes, the base application with majority features functioning, further documentation of implementation, additions to the literature review, and evidence of testing.

### Main Submission

Due on the 26th of April 2024, the main submission is the final milestone and includes everything from milestones 1 and 2, the final application, and an evaluation of the project. Between milestone 2 and the final milestone, changes to the application are to be expected, and suitable documentation is provided where needed.

Alongside the Gantt chart, Trello has been used to keep track of what tasks need completion, what is in process, and what has been done. This changes throughout the development cycle.

|  |
| --- |
| Figure 2 – Trello board |
|  |

# Research

Payroll and management systems are a vital component of a business’s success, protecting the business’ finances but also displaying their commitment to employees. As businesses become increasingly reliant on digital solutions, the role of database management will only grow. This literature review delves into the complexities and challenges surrounding payroll systems and databases.

The following sections will identify relevant sources, technologies and practices within the field of Payroll Systems. I will explore existing literature and applications to improve my understanding of the field, identify areas of improvement and justify the significance of my project.

As technology continues to advance and gain widespread adoption, numerous systems, including payroll management have significantly changed. Delving into the role of a payroll system and exploring the advantages offered to employers and employees provides valuable insight that resonates throughout the project.

Oracle describe payroll software as “an on-premises or cloud-based solution that manages, maintains, and automates payments to employees” (Oracle, 2023). This definition highlights the core functions of a payroll system. Furthermore, Oracle emphasizes that when appropriately configured and integrated payroll software empowers businesses by ensuring they comply with financial regulations and allowing their human resources department to redirect focus towards more tasks, resulting in cost savings. A well-designed and implemented system streamlines operations for both employers and employees, making it an invaluable asset.

When designing and developing my payroll system, there are a multitude of factors to consider. These include the selection of programming languages available, the choice of a robust and efficient database management system, the design of an intuitive and user-friendly interface for presenting the information to end users and the cost of development and maintenance. Evaluating these components is vital to the success and functionality of the payroll system.

The rising popularity of Low Code development platforms shows their potential to change application development. These platforms empower users to create applications with minimal coding knowledge, leveraging a graphical interface instead of traditional programming languages like C#, Java and Python. They offer several advantages, primarily emphasising productivity, efficiency and cost. Making use of these platforms to create my functional prototype decreases time developing and provides a deeper insight into the workings of the system.

As stated by Mendix Technology (2023), low code platforms “boost productivity” by providing a consistent development process, enabling users to rapidly create applications, deducing time to market. Oracle (2023) states that their low-level platform Oracle Apex also helps to reduce costs as it will adapt to new technologies and threats as they emerge, future-proofing applications and streamlining the maintenance and update process.

Additionally, some low-code platforms, such as OutSystems and Microsoft Power Apps are integrating the power of Artificial Intelligence (AI) into their software with the hopes of further enhancing the ease and user-friendliness of their platforms.

## Low Code Platforms

### Mendix

Mendix offers a drag-and-drop modelling environment with the option to write JavaScript code. In June 2023, they integrated AI into their platform, stating that they were the first to enable this type of feature (Mendix Technology, June 2023). Prices vary depending on the use case. Table 1 shows a complete list of the databases (Mendix Technology, 2023) and deployment (Mendix Technology, 2023) options offered by Mendix and indicated in their support documents.

|  |  |
| --- | --- |
| Table 1: Database and deployment options Mendix |  |
| Databases | Deployment |
| MariaDB | Locally |
| Microsoft SQL Server | Mendix Cloud |
| Azure SQL | SAP Business Technology Platform |
| MySQL | Siemens MindSphere |
| Oracle Database | Private Cloud |
| PostgreSQL | Cloud Foundry |
| SAP HANA | Docker and Kubernetes |
|  | Azure |

### Microsoft Power Apps

Power Apps offers a familiar environment for those using Microsoft’s software, such as Word and PowerPoint. Power Apps makes use of a drag-and-drop and preconfigured templates to help create apps quickly. Power Apps also offers AI based “no-code add-ons” (Microsoft , 2020). Microsoft Power Apps is model-driven and allows you to populate the data and have the software design the look for you. Plans start at £16.40 a month and can go above £400 a month depending on the services needed. Table 2 displays the full list of databases (valto, 2023) and the deployment options available.

|  |  |
| --- | --- |
| Table 2: Database and deployment options Microsoft Power Apps |  |
| Databases | Deployment |
| Microsoft Excel workbooks | Microsoft cloud |
| Other Office 365 apps |  |
| SharePoint |  |
| SQL server |  |
| Microsoft Dynamics |  |
| Other CRM databases |  |

### Oracle Apex

Oracle states that APEX is another popular low code platform that allows users to create secure web and mobile applications easily. Oracle APEX allows the user to upload a spreadsheet, and the system will automatically create a table in Oracle Database and an app that can be used to maintain the data. APEX apps can use of Single Sign-on (SSO), Lightweight Directory Access Protocol (LDAP) or Social Login for an extra layer of authorisation/security. Similar to others, APEX offers AI-assisted development that allows developers to use natural language prompts to create anything from SQL queries to entire applications (Oracle, 2023). Additionally, Oracle APEX offers a range of pre-built components and templates to accelerate application development.

APEX empowers users to effortlessly establish REST data sources, providing access to vital data and information. This versatile capability extends to both Representational State Transfer (REST) services or generic JSON data (Oracle, 2023). In 2022, Monica Godoy, a Senior Principal Product Manager for Oracle, wrote an article offering a comprehensive guide on harnessing REST data sources to seamlessly integrate various databases into an APEX application. This flexibility offered by APEX can prove invaluable for organisations and individuals who operate databases other than the default Oracle Database, streamlining their workflows and reducing operational overhead.

In addition to the flexibility offered by the REST data source, APEX offers a range of deployment methods to cater to needs. The APEX deployment page, documented by Oracle in 2023, is a comprehensive resource discussing these deployment options. Like other offerings, APEX facilitates deployment on its proprietary cloud servers, Oracle Cloud. However, this is not the only option; APEX grants its users freedom to host applications either on-premises or within a private cloud environment, regardless of whether the underlying platform is Windows, Unix, or Linux. This level of versatility empowers users to make choices that align precisely with their unique requirements.

### OutSystems

OutSystems allows users to easily create, deploy and manage web and mobile applications. It simplifies the development process by providing a visual interface and offers pre-built templates. Customers can extend their application using standard C# (OutSystems, 2023). OutSystems is free as long as you only create one application; more than one will cost upwards of $1,513 a month (OutSystems, 2023). Table 3 below shows the databases compatible with OutSystems (OutSystems, 2023), along with the possible deployment methods (OutSystems, 2023).

|  |  |
| --- | --- |
| Table 3: Database and deployment options for OutSystems |  |
| Databases | OutSystems Cloud |
| SQL Server (with requirements) | Public Clouds |
| Azure SQL Data (with requirements) | Private cloud |
| Oracle | On-premises |
| DB2 |  |

## Competing Applications

It’s important to research existing applications to gain a deeper understanding of the market

### BambooHR

BambooHR, 2023, positions itself as a time and effort-saving solution for its customers through its software. They proudly brand it as the “Complete HR Platform”, offering comprehensive support for a multitude of HR functions, including:

* HR Data and Reporting
* Hiring and Onboarding
* Employee Experience and Performance
* Payroll, Time, and Benefits
* An Integrated Marketplace

BambooHR’s array of features makes it an ideal choice for businesses seeking a unified system to efficiently manage all HR-related activities however, may be overwhelming for those who just need a payroll system. BambooHR’s adopts a flexible pricing approach tailored to each customer, providing free price quotes on request. They offer two distinct plans, with certain features only accessible through the higher-tier subscription.

### Sage Payroll

Sage Payroll software aims to streamline the payroll process for businesses by providing an affordable, cloud-based system, stating it is suitable for small UK businesses, charities, retail stores, and non-profits (Sage Group, 2023). Sage offers a generous variety of plans, with the first three months free, costing as little as £10 plus VAT a month with a £2 increase per employee up to 150. After the 150, mark there are standard and premium options starting at £20 or £30 plus VAT. Sage allows their customers to upgrade or cancel their subscription at any time. They also several of other applications that might interest their customers.

Sage Payroll allows for the following:

* Payroll management
* Self-serve payslips and P60s
* HMRC real-time information
* Pension enrolment
* Management of employee holidays

### Wave

Wave states on its main page that it offers “Everything you need. Nothing you don’t” (Wave Financial Inc., 2023). Wave has a different software available for purchase:

* **Invoicing**: allowing businesses to send invoices to clients.
* **Payments**: manages how businesses can be paid and shows the payment status.
* **Accounting**: monitors cash flow to help businesses stay organised.
* **Payroll**: easy management of payroll for the business.

Wave Payroll offers the ability to pay employees and contractors, an automatic payroll journal, seamless integration with their other software, automatic tax payments, and direct deposits. Wave Payroll starts at $20 with a plus $6 per active employee, Wave also offers a 30-day trial for interested users.

## Data Protection and Privacy

Data protection is a highly important topic in today’s digital world. My project places a significant emphasis on safeguarding personal information. Commitment to data protection isn’t merely a legal requirement but a moral and ethical responsibility. As development begins, it’s imperative to consider and uphold the rights and privacy of individuals, particularly when handling sensitive payroll data. During the development application, no real-world data will be utilised.

The United Kingdom has several data protection and privacy legislation in place to protect the rights of individuals and regulate the handling of personal data. Some of the legislations that will apply to the application include:

* Data Protection Act 2018 (UK GDPR): The UK’s primary data protection legislation. It incorporates the European Union’s General Data Protection Regulations (GDPR). This outlines the rules and regulations for processing of personal data by organisations, businesses and the government (UK Government, 2023).
* Privacy and Electronic Communications Regulations (PECR): The PECR works alongside the UK GDPR legislation mentioned above it provides specific privacy rights concerning online or electronic communication (ico, 2023). This piece of legislation is relevant to the project as, my payroll system incorporates an automated email service that will be sending sensitive data.

This legislation is subject to change/update, and because of this, ensuring an understanding of the latest legislation during the development of my payroll system is highly important.

Considering the criteria set by UK legislation, implementing a robust data protection framework ensures I adhere to this. Here are some key principles and measures incorporated:

* **Data Encryption**: Sensitive data, such as employee payroll information, will be encrypted whilst stored in the database and when transmitted to employees. This ensures that even in the unlikely event of a data breach, the information gained is unreadable and secure.
* **Access Control**: Access to the application will be strictly controlled, and only authorised personnel will have access to the system. Access levels will be appropriately tailored/restricted based on the user’s role and responsibilities within the business.
* **Data Minimisation**: The application collects and stores only the minimum amount of data necessary for it to operate correctly. All users will be able to see all of their personal data.
* **Data Retention Policy**: A clear data retention policy, ensures that data is only retained for the necessary period and then securely deleted or anonymised. The organisations using my payroll system may have their own data retention policy that will override mine; however, by creating one to ship with the application, I am ensuring there always is one in place.
* **User Consent**: No data will be collected without explicit consent from the user and no data will ever be sold to third parties.

By incorporating these measures and maintaining a strong commitment to data protection, the payroll system not only meets the legal requirements but also upholds the trust and confidence of those whose data is stored and used by the application. Below is an example of the privacy policy.

|  |
| --- |
| **Privacy Policy**  **Data Collection and Usage**   * Our application collects and stores personal information such as usernames, passwords, and employee details for the purpose of payroll management. * This information is used solely for internal purposes and will not be shared with third parties without explicit consent.   **Data** **Security**   * We are committed to protecting the security of your data and have implemented measures to safeguard against unauthorized access, disclosure, alteration, or destruction. * Sensitive data is stored securely using encryption and access controls to prevent unauthorized access.   **Data Retention**   * We retain personal information for as long as necessary to fulfil the purposes outlined in this privacy policy unless a longer retention period is required or permitted by law. * When data is no longer needed, it will be securely deleted or anonymized to prevent identification.   **User Rights**   * Users have the right to access, update, or delete their personal information stored in our system. * To exercise these rights, users can their administrators.   **Changes to Privacy Policy**   * We reserve the right to update or modify this privacy policy at any time. Any changes will be reflected on this page, and users will be notified of significant updates. * Continued use of the application after the changes are made indicates acceptance of the updated privacy policy. |

# System Design and Implementation

## Prototyping

Prototyping is useful as it allows the creation of a working model or representation of the payroll system. This prototype helps visualise, test and refine the idea before full-scale development begins.

For prototype and final application design, ‘Justinmind’ will be utilised. ‘Justinmind’ is a versatile tool that simplifies the process of crafting pages and allows experimentation with a range of user interface and user experience elements. Oracle APEX is the low code platform of choice for prototype development.

### Dashboard

The dashboard is the landing page of my application. Here, users are able to quickly access all other pages, see how many hours they are working in the current week, see how many hours they have available to book off, and see any updates to the software.

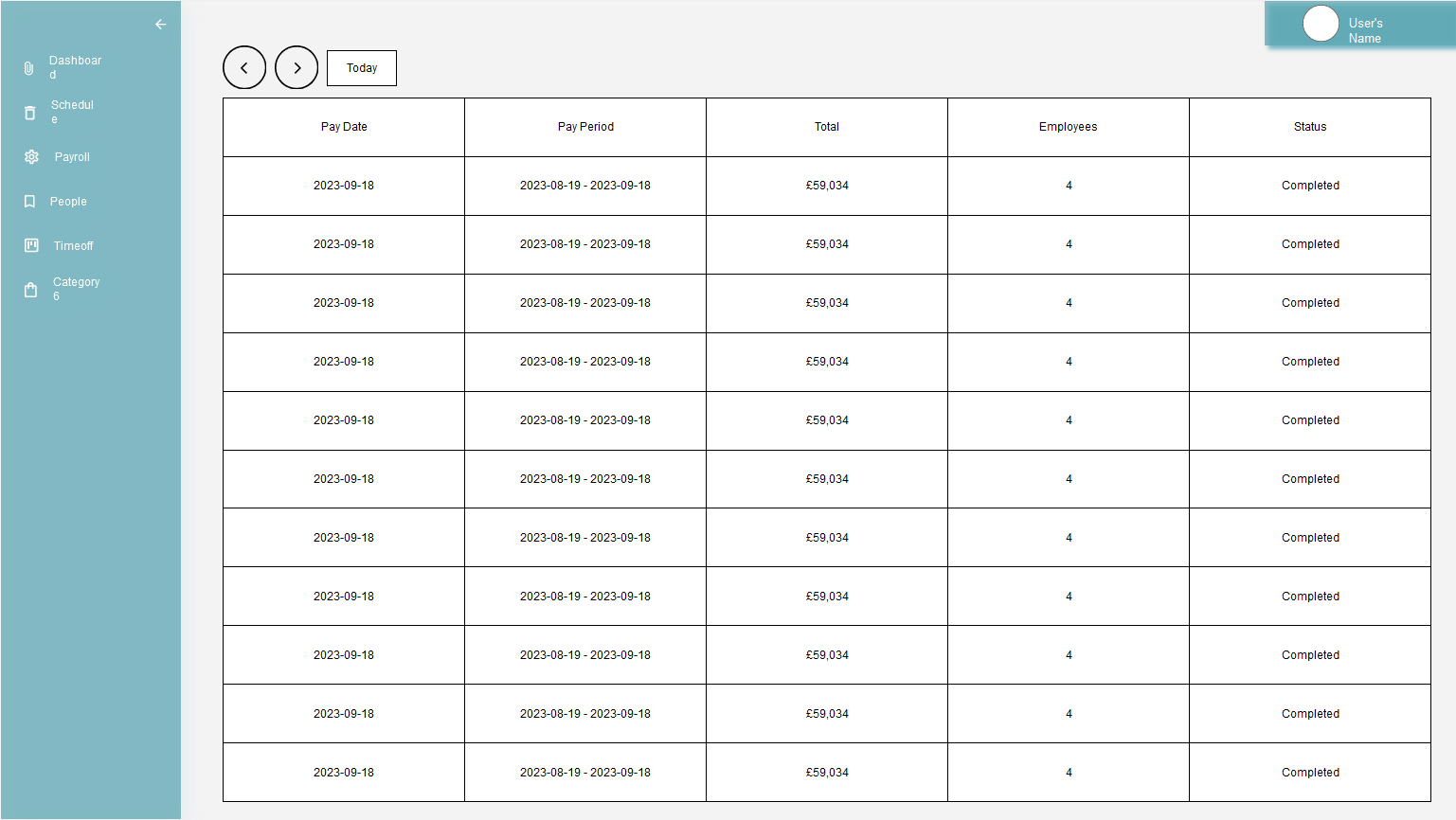
### Schedule

The schedule shows the day’s users are working along with any days booked off and what day they will be paid that month. Users will only be able to see their own schedules.



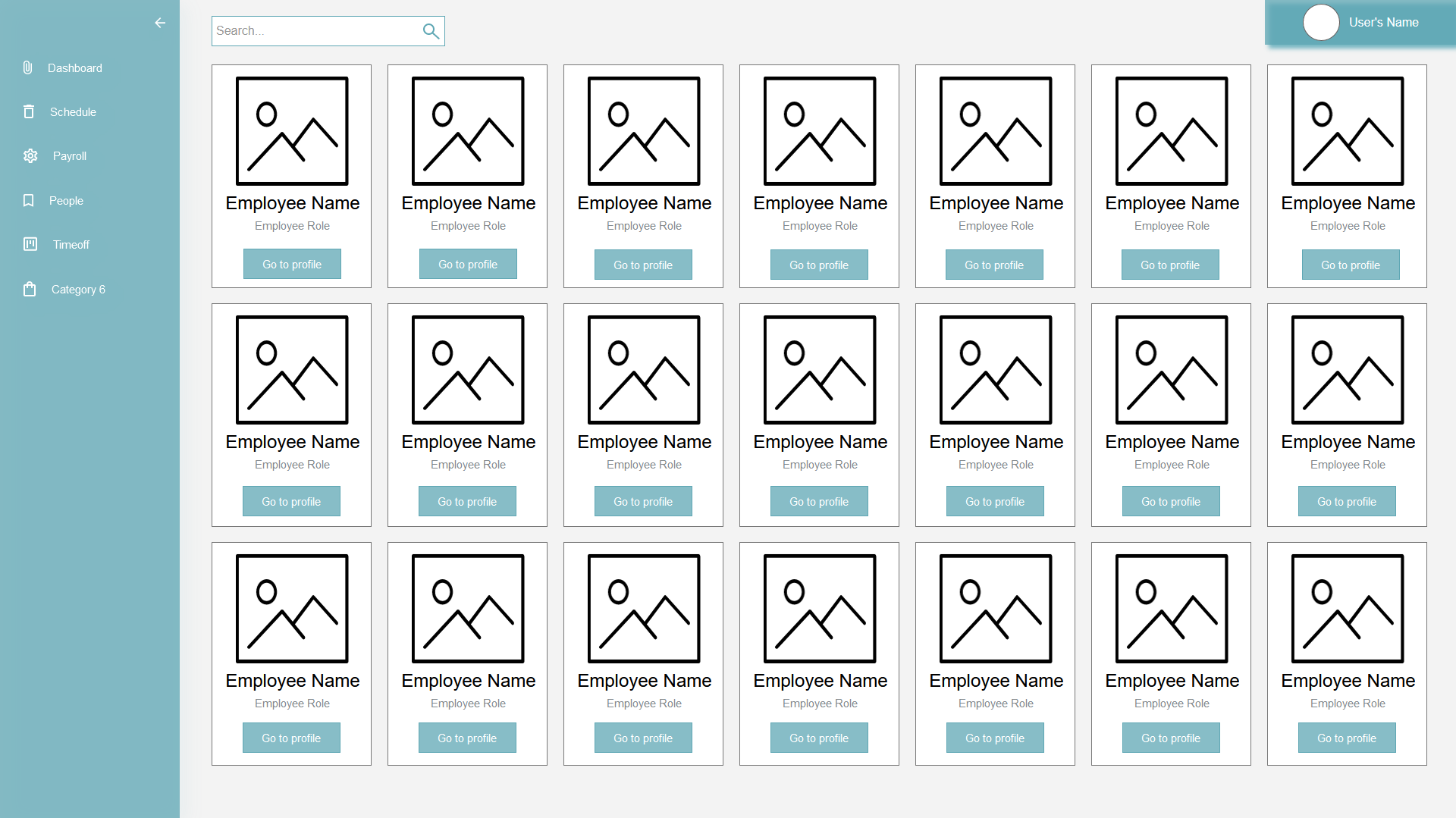
### Payroll

This page is only accessible to those with the correct permissions, allowing users to view information and act as a log/journal for payrolls.



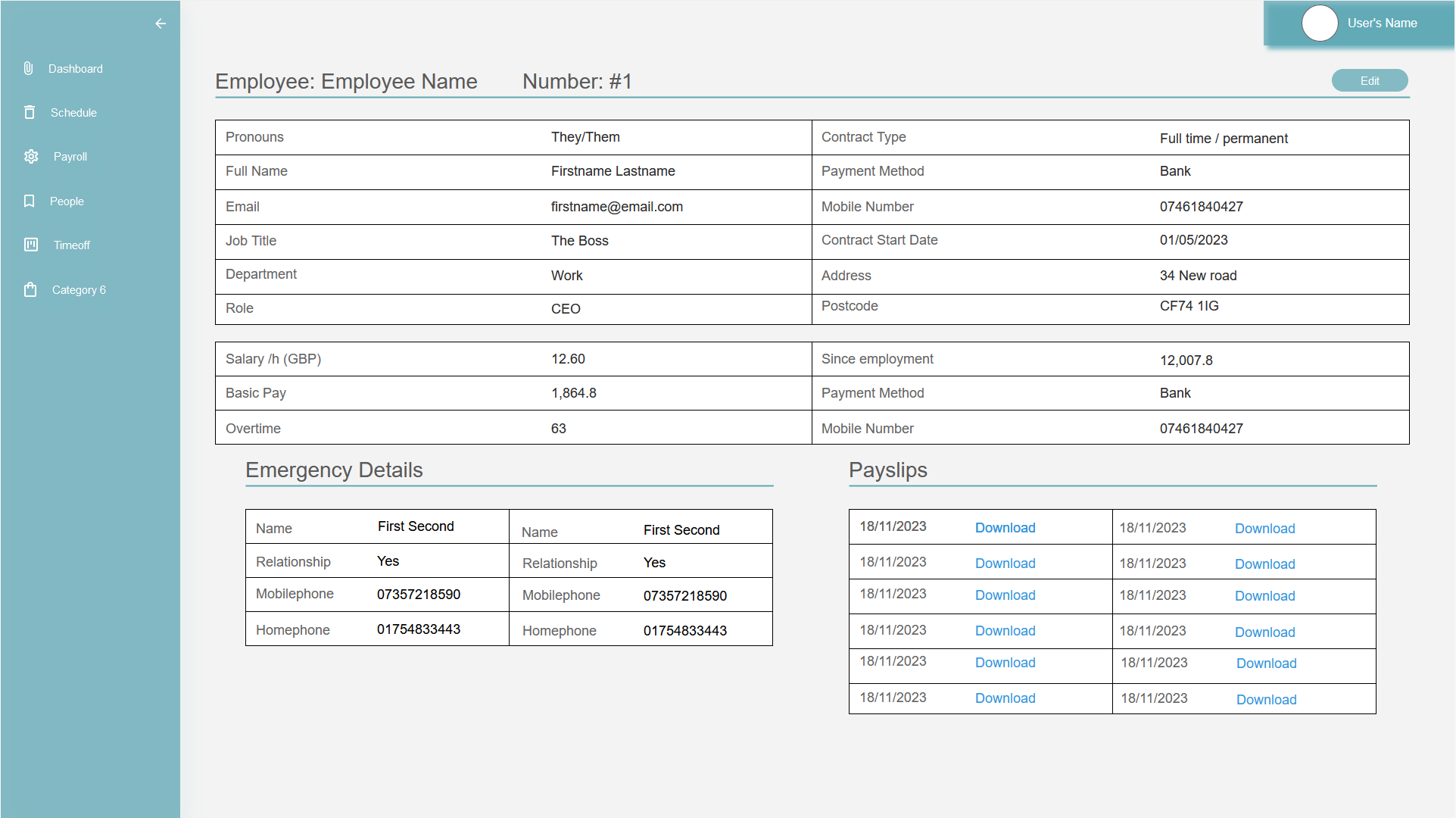
### People

The people page allows those with access to search and view other employee information along with a way to modify data.



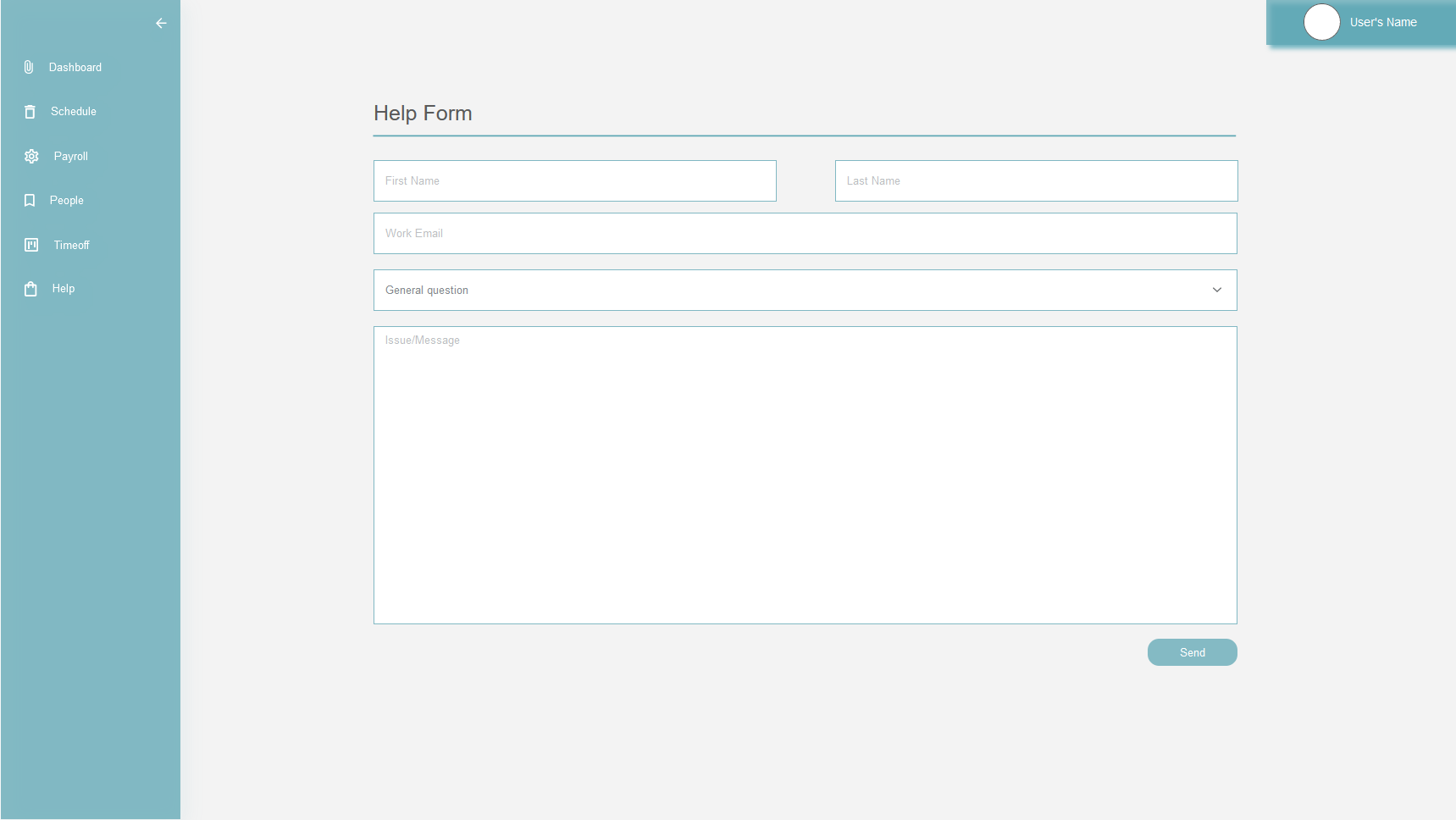
### Profile

The unique profile for each employee displays personal information therefor is only be accessible by the employees themselves and those with correct permissions. Here, users are able to view and edit the personal information of the employee. Also available for download on this page are previous payslips.



### Help

A simple help form to reach out to the business support team. Users receive responses and continue the conversation over email past this point.



## Table Design

While designing the payroll system’s database, the primary objective was to create a robust and scalable structure that efficiently handles employee information and payroll records. This section discusses the database’s attributes, relationships and normalisation.

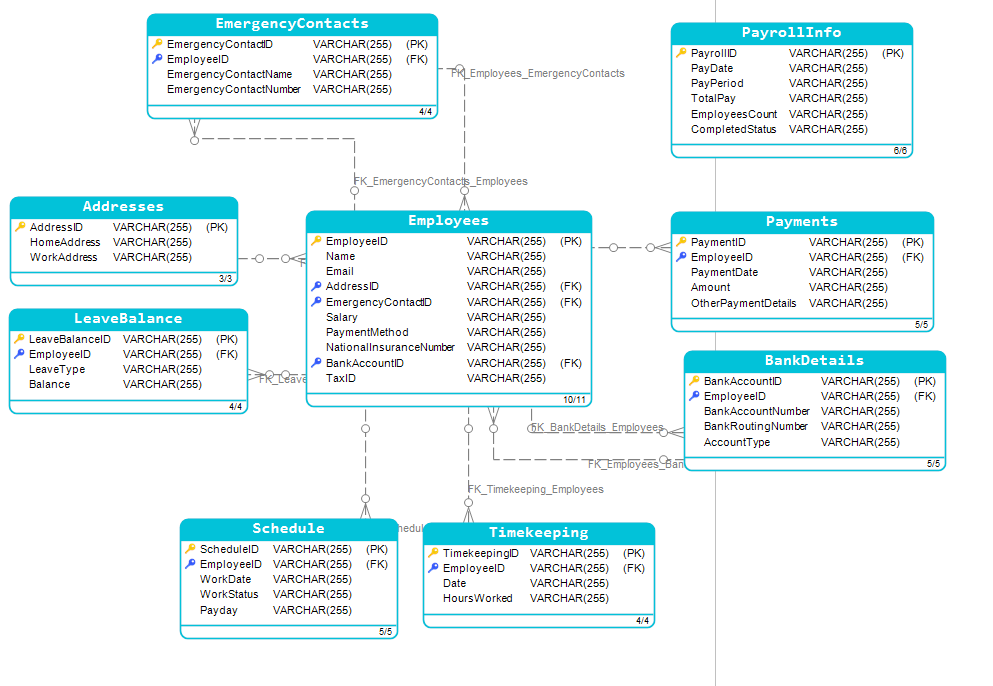
### Entity Relationship Diagram (ERD)

The ERD serves as a foundation for the database structure, a blueprint of sorts. It illustrates the connected entities and their attributes, detailing what data should be stored within each table.

#### Entities and Attributes

The ‘Employees’ table acts as the centre of the database; everything is connected to this table. Each ‘Employee’ entity will have the following attributes: ID (Primary Key), Name, Email, Address, Emergency Contact, Salary, Payment information, National Insurance Number, and Tax information. Many of the attributes will be stored in the other tables and connected via a foreign key.

The employee’s emergency contact is stored within the ‘Emergency Contacts’ table, and it holds the primary key ID and a foreign key linking it to the ‘Employees’ table via the Employee’s ID. Separate from these keys, the table stores the emergency contact name and phone number. Each employee can have multiple emergency contacts and, there is no limit on how many employees one can be the emergency contact for.



#### Relationships

Establishing the correct relationships between database tables helps ensure data integrity, consistency and more. It supports good database design principles and is important for several reasons:

1. Data integrity – Relationships ensure that data is accurate and reliable.
2. Consistency – Without proper relationships data may be duplicated or inconsistent, leading to confusion and errors. Relationships will help maintain consistency.
3. Normalisation – Relationships support the normalisation process. Normalisation is discussed further in the section below.
4. Enforces Rules – To ensure certain conditions or requirements are met when data is inserted, updated or deleted, suitable constraints are used.
5. Scalability – Well-designed databases with relationships are more scalable.

|  |  |  |
| --- | --- | --- |
| Table 1 name | Table 2 name | Relationship |
| Employees | Payments | One to many |
| Employees | BankDetails | One to one |
| Employees | EmergencyContacts | One to many |
| Employees | Addresses | One to many |
| Employees |  |  |
|  |  |  |
|  |  |  |

#### Normalisation

Normalisation is the process performed on a database to organise the attributes and reduce data redundancy (GeeksforGeeks, 2023). This is important because repeating data can increase file size and create conflicting data.

Performing normalisation involves breaking down larger tables into smaller, related tables. The process can involve several stages, known as normal forms (NF). Each normal form has a set of rules and a purpose, and each subsequent normal form builds upon the last.

## Oracle APEX Prototype

As part of my journey to create my payroll management system, the initial prototype was developed using Oracle APEX. This prototype served as a dynamic and functional representation of the app, showing the fundamental features and interactions.

Leveraging the power interface capabilities of Oracle APEX allows testing of different user interfaces and page navigation options. Oracle APEX’s easy integration with databases allowed for real-time data manipulation and retrieval, offering insight into the system’s backend functionality.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

# Application Development

Building upon the Oracle APEX prototype, the decision to use Java and JavaFX is a strategic move towards greater customisation, flexibility and control over the application’s development.

## Advantages of Java and JavaFX

1. Java has a “write once, run anywhere” philosophy ensuring that the payroll system will be platform-independent. Java achieves this through the use of the JVM or Java Virtual Machine (GeeksforGeeks, 2023).
2. There is an extensive collection of libraries and frameworks available for Java.
3. The Java ecosystem is scalable and aligns well with my project requirements.

## Technical Explanation

### Libraries and Tools

#### Version control - Git and GitHub

During development, I will leverage Git and GitHub for source control. This tool facilitates collaboration for both individuals and teams but also enhances project organisation. GitHub’s features streamline code management, version control, and issue tracking to ensure a well-structured and organised development process. Along with all of these benefits, Git and GitHub are used together to ensure there is an off-site backup of my application.

#### IDE - IntelliJ

IntelliJ is an IDE (Integrated Development Environment) created by JetBrains for Java, Kotlin, Groovy and other languages. Beyond its easy-to-use interface, IntelliJ comes packaged with a suite of useful tools and supports customisation through the use of plugins to further improve the experience.

#### Dependency management - Apache Maven

Maven is a tool that is used to manage and build Java projects (The Apache Software Foundation, 2024). Maven’s sophisticated dependency management system, incorporating external dependencies or libraries becomes a streamlined process. This tool significantly simplifies project structuring and ensures that when moving between computers the correct dependencies are automatically downloaded.

#### UI build tool - SceneBuilder

Responsible for the development of an intuitive and visually appealing user interface JavaFX and SceneBuilder have been employed. Available both as a standalone application or a plugin for IntelliJ, SceneBuilder provides a user-friendly drag-and-drop environment. This tool is instrumental in enhancing the speed and precision of my UI development.

#### Database - Oracle 21c Database

Maintaining a commitment to stability and consistency, I have opted to use the Oracle 21c Database for the final iteration of my application. The Oracle database can easily be integrated with Java and IntelliJ. This choice allows me to seamlessly transition from my APEX prototype, utilising the same database design.

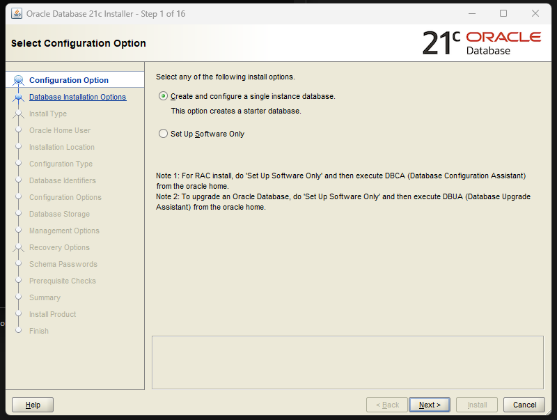
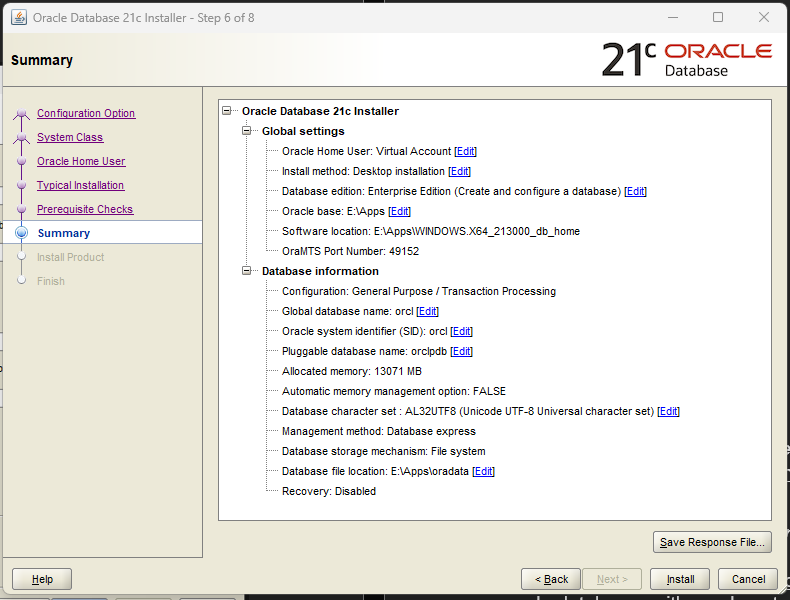
#### Database viewer - SQL Developer

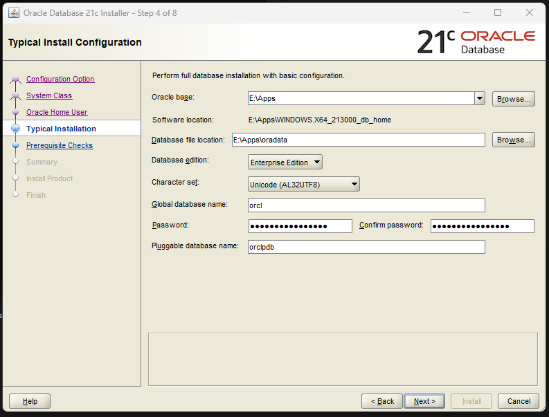
Oracle offers SQL Developer as an application to easily view and manage Oracle databases. This tool provides real-time updates, granting me the ability to monitor changes to the database instantly. This capability enhances the efficiency of testing and development, offering a transparent view of database modification as they occur.

### Database setup

#### Oracle 21c Database installation

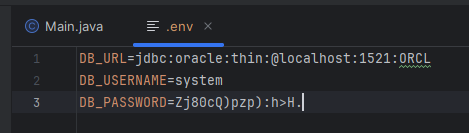
The Oracle 21c database was set up following the comprehensive installation instructions provided by Oracle. Initially, a local host configuration is used; however, the flexibility of Oracle 21c and Java enables seamless migration to a different environment by adjusting the database URL accordingly. This adaptability ensures the scalability of my project beyond the confines of local hosting, possibly allowing a separate server and client application.





#### Environment File and Data Security

Working towards safeguarding sensitive data, my project employs a “.env.” or environment file. This file which is crucial for storing and accessing unique data, remains absent from the GitHub repository to preserve data integrity and security. On the application’s first launch, this environment file is generated and ready to be populated by the user. To further enhance security sensitive data will be encrypted before storing in the database.



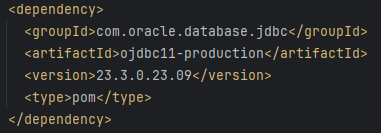
#### Security Measures for Database Access

The database is intentionally restricted from external accessibility, ensuring it remains exclusively available within the confines of the local machine. This precautionary measure mitigates potential security risks associated with external access during development. Additionally, upon completion of the project, the database will be closed, further protecting the host machine.

### Implementation

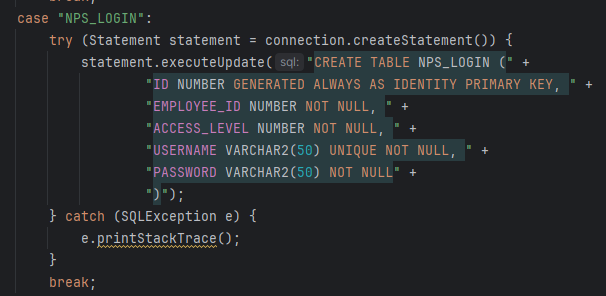
Exploring the practical implementation of my payroll system is the purpose of this section. I’ll discuss key features currently implemented, showcasing code snippets for this.

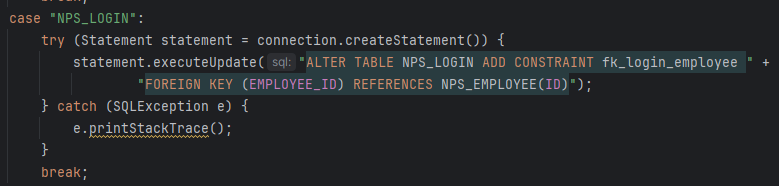
Before interfacing with the database, a crucial step is to establish the connection to the database. The Oracle JDBC (Java Database Connectivity) driver handles this task. Maven facilitates the inclusion of necessary dependencies. The specific code for this dependency can be found below and on the Maven repository (<https://mvnrepository.com/artifact/com.oracle.database.jdbc/ojdbc11/23.3.0.23.09>).

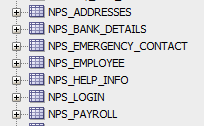
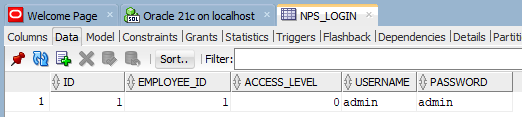


A critical task is completed on the first boot – ensuring the presence of the required database tables. Should these tables not exist, the application creates them. First, the tables are generated as needed and subsequently fine-tuned to incorporate essential constraints. The following code snippet shows the creation process for the “NPS\_LOGIN” tables. Currently, a console message is outputted notifying me if each table currently exists or has been successfully created.

A similar process is taken for every table in the payroll system, ensuring a consistent and dynamic approach to database management. This methodology ensures that each table undergoes the necessary checks, creation and modification steps.

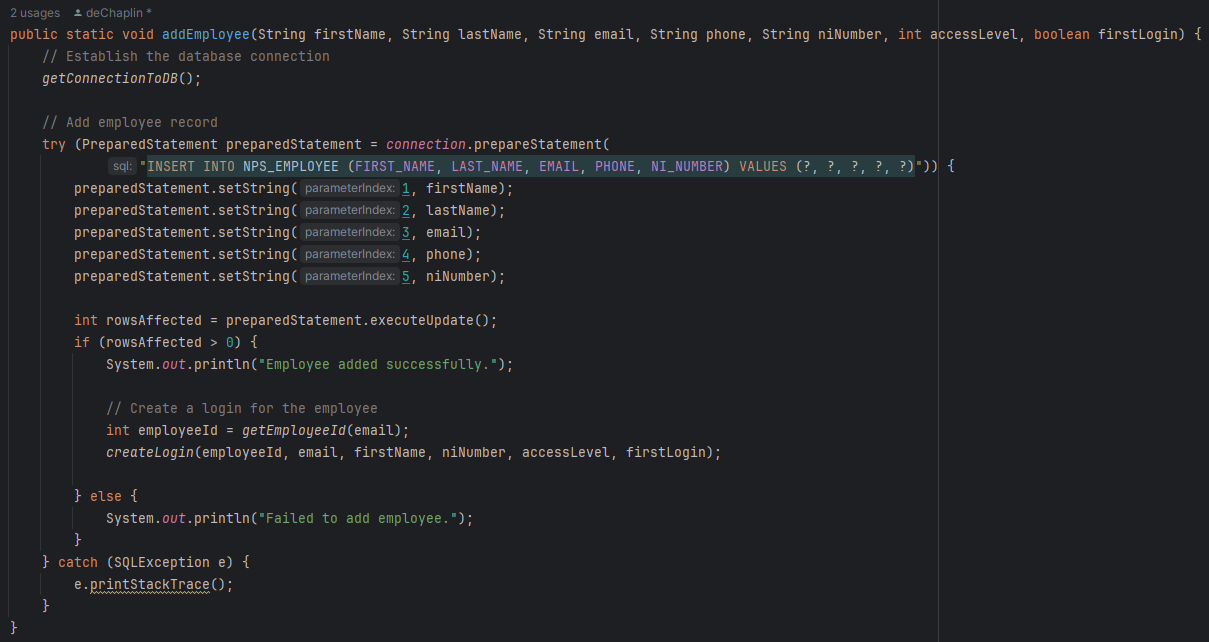


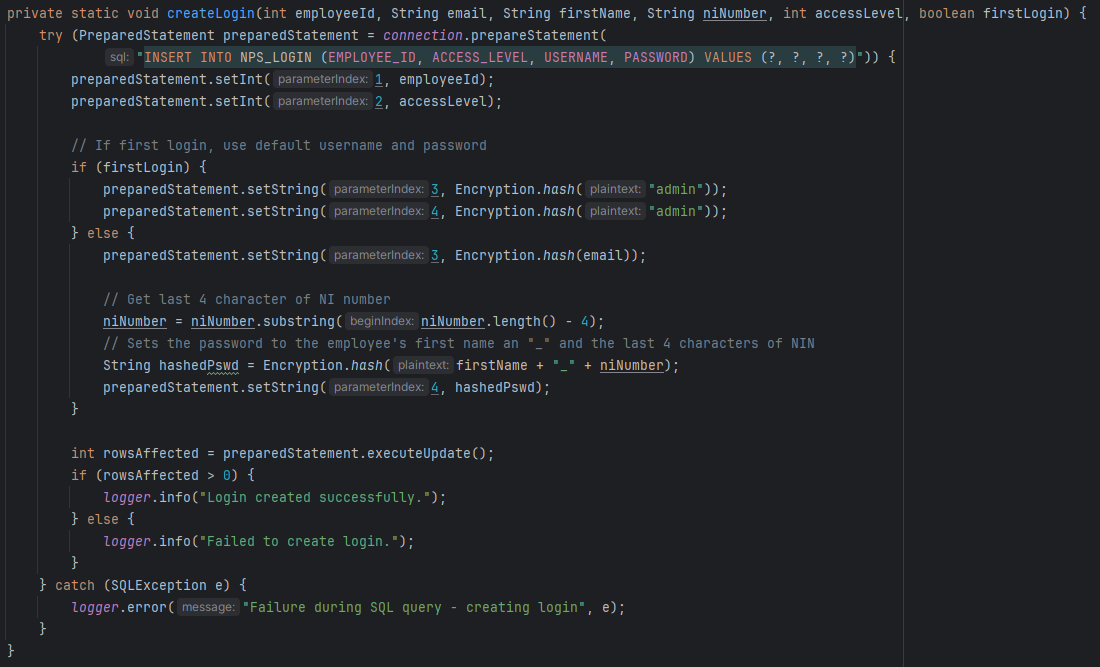


As shown below, tables are accurately created within the database, providing the foundation for the application to store and manipulate information.

As demonstrated above, creating database tables is pivotel for the payroll system. The process extends seamlessly into the “addEmployee” method, a key function for introducing new employees to the system. This method includes essential employee details, including, name, email, phone number, national insurance number, required access level, and a boolean parameter denoting first login creation status.

When calling the “addEmployee” method, the specified parameters are required for the generation of both the employee’s profile and login credentials. Notably, the first login parameter will remain false, barring a single scenario – creating the initial admin account when no other accounts exist in the application. Login credentials are automatically created using the “createLogin” method. Throughout the application SQL queries will utilise prepared statements, this will improve performance and help prevent SQL injection vulnerabilities.



The “createLogin” method plays an important role in generating secure login credentials for employees. Making use of and combining the provided email, first name and last four characters of the national insurance number, this approach ensures a balance between security and user convenience. The chosen method allows for communicating login information to employees through a simple message without the need to transmit passwords in plain text, prioritising data security. Below is a code snippet of this.

Each user has the capability to access and modify their personal information through the profile page. Upon switching to edit mode, editable text fields are visually distinguished by a green highlight, ensuring easy identification by for the user. In case of an input that fails to pass the systems validation, an error prompt is displayed, and the problematic text field is highlighted in red for correction. Additionally, each user can conveniently download their last six payslips directly from the profile page. Below, each step is displayed along with an example input validation check.

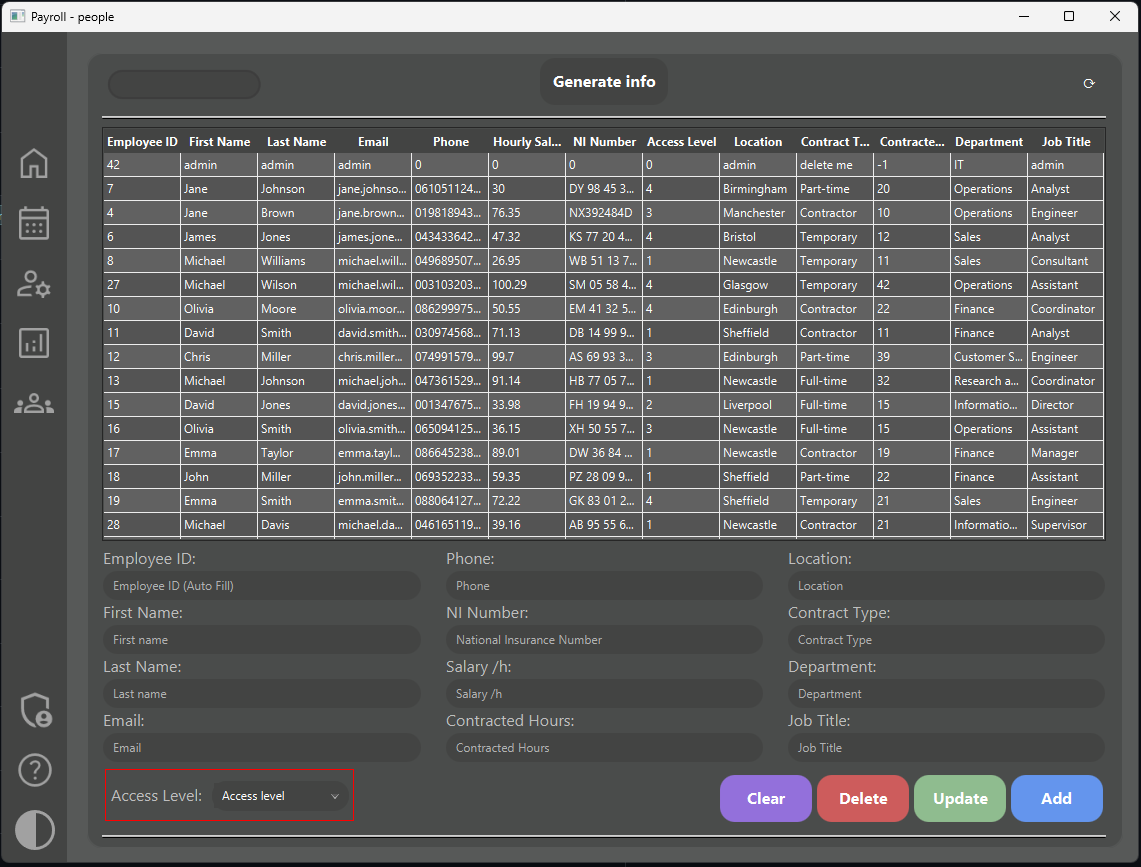
|  |
| --- |
| Employee page in edit mode |
|  |
| Employee page when a text field fails input validation |
|  |
| Payslips available to download |
|  |
| Example input validation for the National Insurance number |
|  |

Throughout the system, input validation is implemented to ensure data integrity and accuracy, Initially, checks are performed to ensure that mandatory fields are filled, preventing submission of empty data. Subsequently, Java regular expressions, a string of characters with a specific pattern (Refsnes Data, 2024), are employed to validate each input against a predefined criteria. Valid data matching the specified pattern is accepted, while any deviation will result in an error message to prompt the user for correction.

Additionally, many pages come equipped with a search function which allows the user to easily and quickly find specific data in a table view. As an example, in the employee page a user can search for an employee record using their ID, first name, second name or email and, the table view dynamically updates as the user types.

|  |
| --- |
|  |
|  |
|  |

The privileges to add, delete, and update employees is restricted to users with the appropriate access levels. The user interface has been designed to allow the user to view, edit, delete and add records all on a single page. When a record is selected in the table view, the corresponding data populates the text fields below. This streamlines the process of modifying records without the need for multiple different pages for editing, deleting and adding.



Clears the text fields.

Delete the currently selected record.

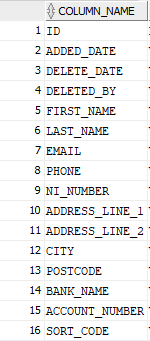
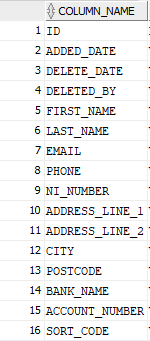
Updates the record with any changes made in the text fields.

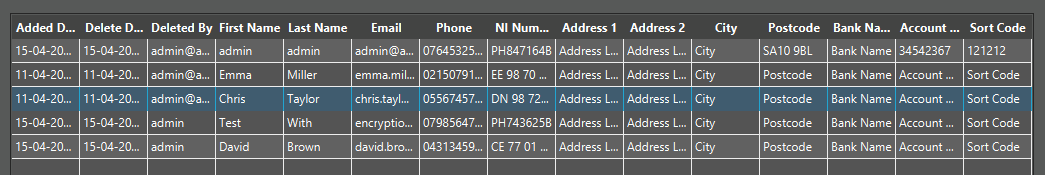
Adds a new record using the information in the text field.



Upon successful deletion of a user’s record, the system initiates an automatic process to safeguard relevant data, anticipating potential future requirements. This involves transferring information to an additional table named “NPS\_DELETED\_USERS”. This practice ensures that if communication is required with the former employee, it is possible and provides accountability by preserving a record of who deleted the record.

ICO data retention guidelines stipulate that organisations may retain data as long as necessary, provided they can justify the retention (Information Commissioner's Office, 2024). The application adopts a selective approach, retaining only essential data following an employee deletion. By following these guidelines, the application adheres to UK GDPR. This personal information can be seen below along with how it can be viewed in the admin panel.

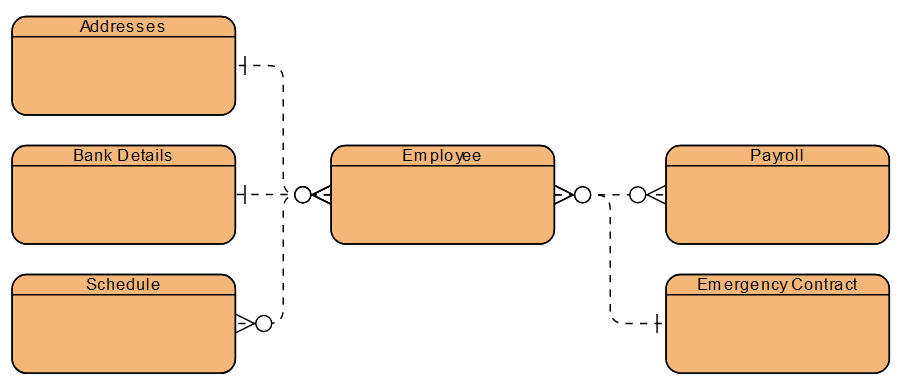




As development progressed, the realisation dawned that storing monthly payroll records for every employee presented a complexity beyond the initial assessment. In response, investing time into researching various options to identify the most suitable approach for my system. Two solutions were presented online: creating a new table for each month (discouraged for its complexity and potential inefficiency) and utilising a single table with a dedicated field for the month.

In a move to mitigate redundant data and prioritise scalability, a single-table approach was adopted. This decision prompted necessary modification to the existing tables to ensure integration of monthly payroll records whilst adhering to best practices in database design. The modified tables can be seen below.

|  |  |  |
| --- | --- | --- |
| NPS\_PAYROLL | NPS\_EMPLOYEE | NPS\_ADDRESSES |
|  |  |  |
| NPS\_BANK\_DETAILS | NPS\_SCHEDULE\_(0,1 ,2, 3) | NPS\_EMERGENCY\_CONTACT |
|  |  |  |



When viewing payroll data, the user is presented with two table views; the top for the general monthly information and the bottom is empty until a selection has been made in the top; it then shows detailed information for that month’s payroll.

Upon accessing payroll data, users are greeted with a dual-table interface. The upper table displays an overview of the monthly payroll information. Meanwhile, the lower table initially remains unpopulated until a selection is made in the update table. Once a selection is made in the top table, the lower dynamically populates, revealing detailed information about the selected month’s payroll. This design intends to streamline the user experience and contextually provide detailed information.

|  |  |
| --- | --- |
|  |  |

The system automates the calculations for pay, pension contributions, taxes, and other relevant factors whenever the schedule is updated. Any changes are instantly reflected in the payroll table view. These calculations are executed using a map to store individual employee details. The system then performs calculations in the required sequence. Once completed, the final data is stored in the database.

|  |
| --- |
| Total hours worked, pension contribution percentage, overtime and current gross pay |
|  |
| Pension contribution amount, net pay |
|  |
| Tax calculations using the current net pay to see what band the user falls under |
|  |
| Updating the database |
|  |

With the implementation of access control, the system operates with distinct access level options. These access levels dictate the privileges granted to users, ensuring that only individuals with the appropriate permissions can access certain features, such as the admin panel or the ability to modify schedule data.

Another security feature integrated into the system is encryption, ensuring the protection of sensitive data such as emails, names, and addresses before they are stored in the database. Additionally, login information is protected through hashing, a cryptographic technique that enhances security by rendering data exceedingly difficult to decrypt.

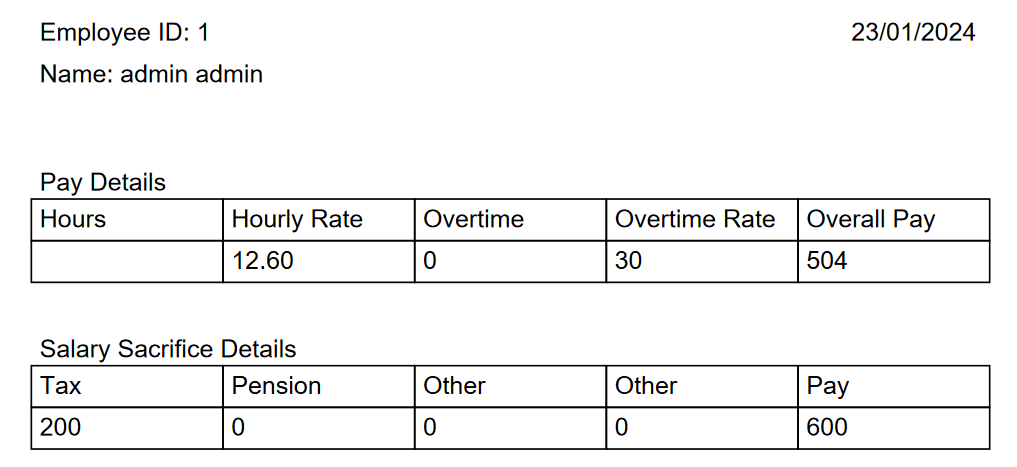
The chosen encryption method is the Advanced Encryption Standard (AES) algorithm. This algorithm uses a predefined key to encrypt and decrypt data. During encryption, the cipher class is initialised with the AES algorithm, and the key is used to perform encryption on the data. The results are strings encoded using Base64 to ensure compatibility with storing the data.

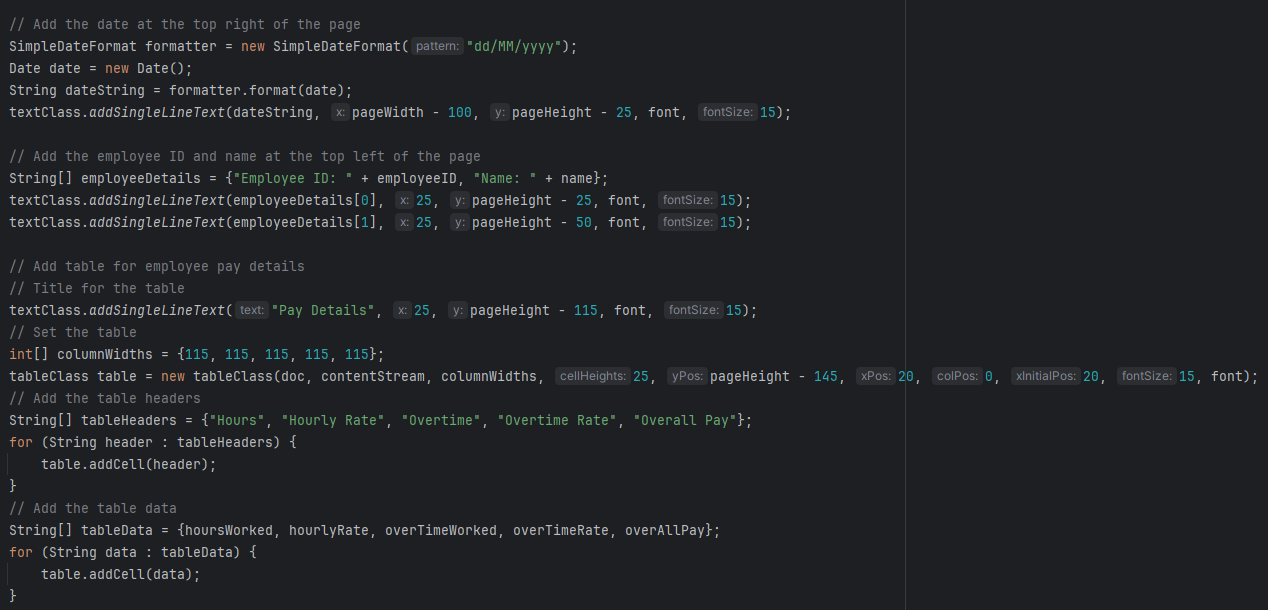
For hashing, the SHA-256 algorithm is used to generate a unique hash value for the provided string. The has value, represented by a hexadecimal string that acts as a digital fingerprint of the original string. The hashing process ensures data integrity, as even small changes to the inputted data results in a significantly different hash value.

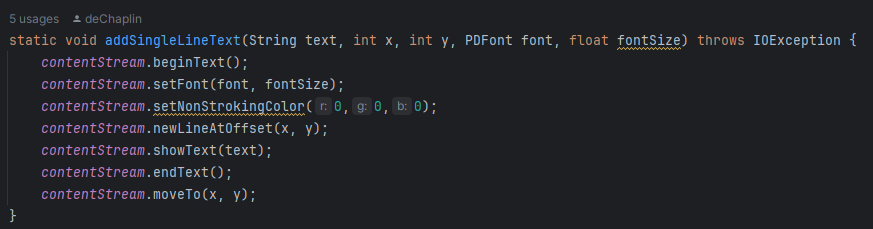
By combining both encryption and hashing techniques, the system safeguards sensitive data. Below are examples of data after encryption and hashing along with the relevant method.

|  |  |
| --- | --- |
| Encrypted data and the “encryptString” method | |
|  |  |
| The “decryptString” method used to decrypt data ready to be displayed | |
|  | |
| The hashed login data and hashing method | |
|  |  |

A core feature of the payroll system is the creation of personalised payslips. To achieve this task, PDFBox, a library developed by Apache, has been used. PDFBox offers a robust framework for generating and designing PDFs programmatically, aligning perfectly with the requirements of my use case. Payslips are created before payday and sent to the employee via email. Below is an example of the PDF created along with a snippet of code.



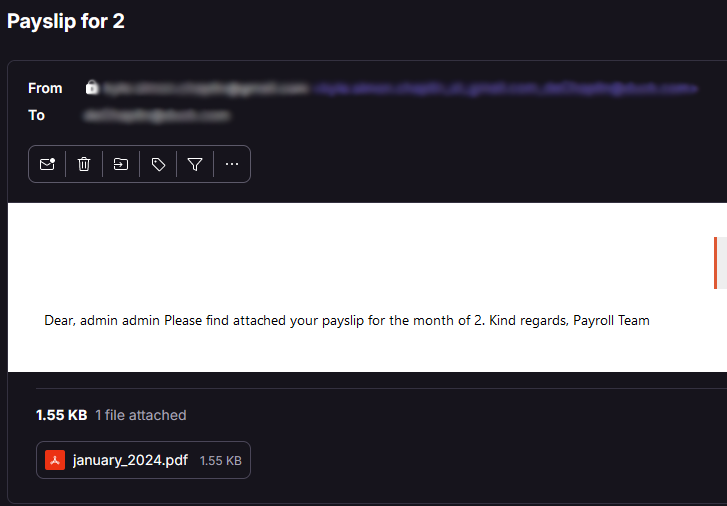


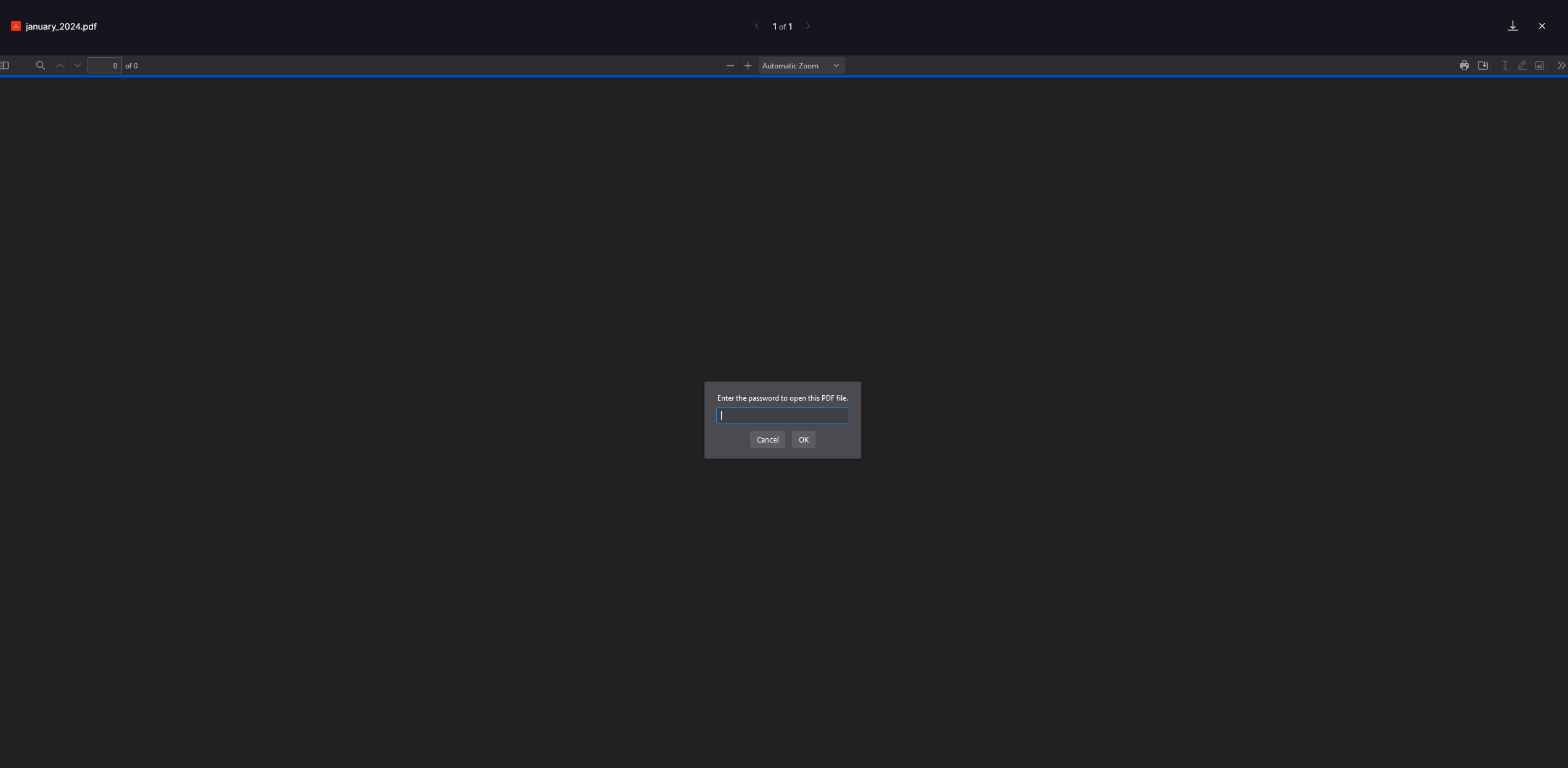
The code above is a snippet of the process of generating PDFs within the system. Following the retrieval and formatting of the current date, data is pulled from the database and formatted to the document using the “addSingleLineText” from the “textClass” class.

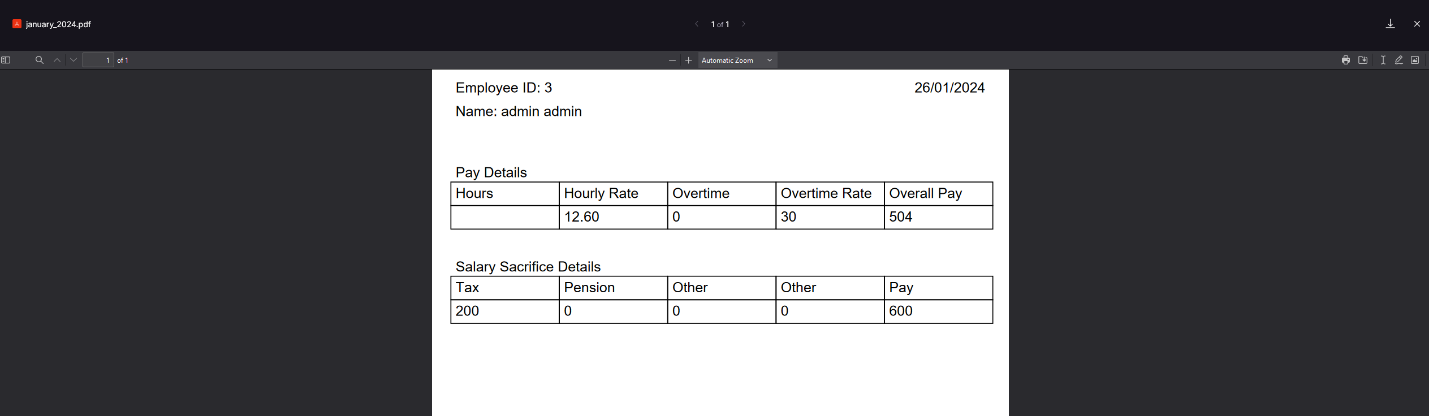
Another essential feature of the system is the automatic generation of emails, each containing a unique PDF payslip. For this purpose, Java Mail, along with Google SMTP, has been used. Creating a timer that, when triggered, an email be sent to the recipients on the chosen time and date. Below are screenshots showcasing code for key aspects of the email automation process.

|  |
| --- |
| Creating timer schedule |
|  |
| Setting the desired time and date to send the email and calling all methods |
|  |
| Setting up Google server properties |
|  |
| Drafting the email |
|  |
| Sending the email |
|  |
| Email received |
|  |

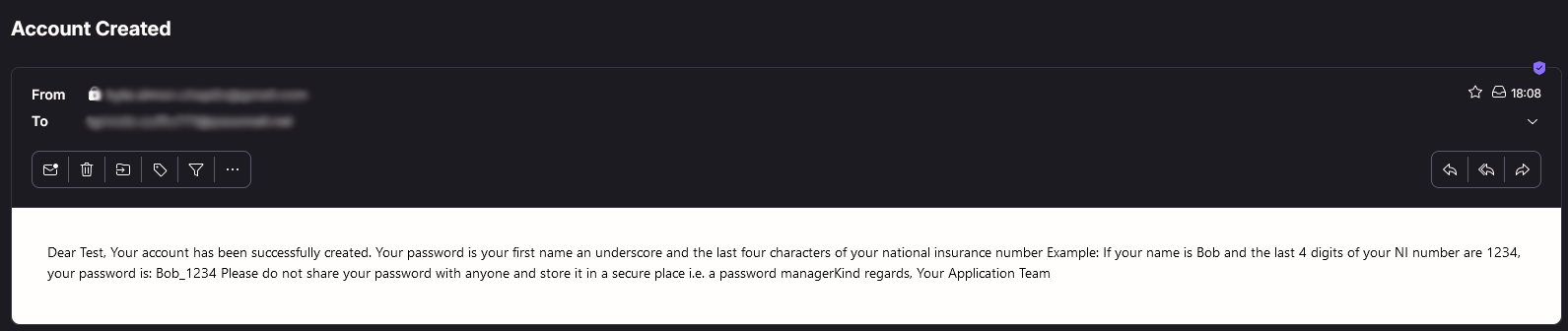
Furthering the development of the email system, the PDFs from earlier is now automatically password-protected, using the last four characters of the employee’s national insurance number, and attached to the email.







Upon registration, a new user receives an automatic email notification to confirm their account activation. This email provides account details, including an explanation of their password. AN example of this notification email is below.



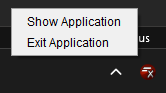
To enhance the user experience and customisation, a user-friendly dark/light mode switch has been incorporated. Leveraging JavaFX’s support for Cascading Style Sheets (CSS), two distinct CSS files, named dark.css and light.css, have been created. The “ThemeManager” class is used to store, retrieve, and toggle between the chosen colour mode. A listener, activated upon pressing the toggle mode button, dynamically updates the selected stylesheet, providing users with the flexibility to change the application appearance. Below are code snippets for the “ThemeManager” and the listener.

|  |
| --- |
|  |
|  |

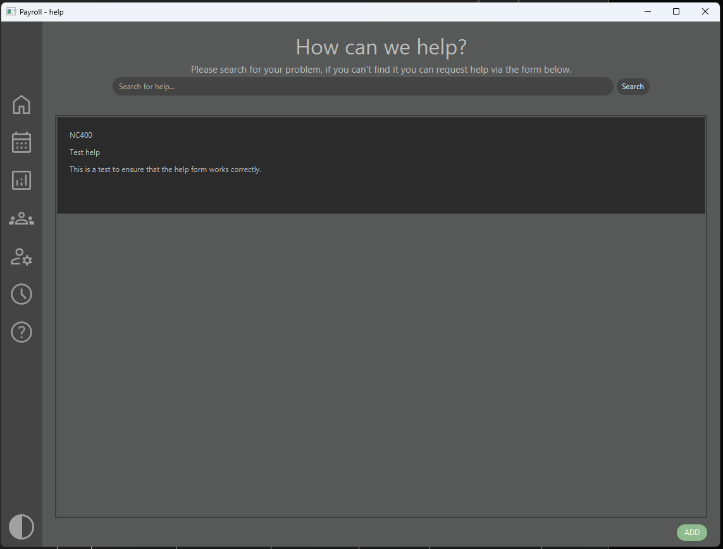
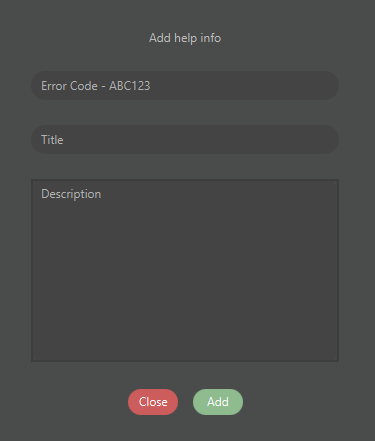
Below is a comparison between the light and dark mode offered. The corresponding CSS files are accessible through my GitHub repository - <https://github.com/deChaplin/JavaFX-light-dark-mode/tree/main>.

|  |  |
| --- | --- |
|  |  |

To prevent accidental closure of the application, the FXTrayIcon library has been incorporated, available at <https://github.com/dustinkredmond/FXTrayIcon>. This library facilitates the creation of a cross-platform compatible tray icon, ensuring the application can run discreetly in the background. With this implementation, the creation of automatic tasks becomes more straightforward.



In addition to the themes available, development has begun on a simple help page to provide valuable information in a user-friendly manner. Access control will be added to ensure that only those with the correct knowledge can contribute to the page, ensuring all information is correct and relevant. The help page will continue to evolve, with plans to implement the access control, search feature, and further collaboration by creating a ‘suggest’ form where users can suggest modifications or additions.



### Code improvements

During the development phase, navigating the “DatabaseController” class to locate specific methods became challenging. This class was tasked with managing all interactions with the database, including encryption, data insertions, retrieval, and other operations. In an effort to improve readability, maintainability, and scalability, I refactored and reorganised this class, adhering to the Single Responsibility Principle (SRP). By following SRP, the class was broken up into smaller, more specialised classes, each with a dedicated responsibility (Martin, 2009). For instance, one class handles database reading operations and another focuses on writing to the database.

During testing and using the application I noticed a distinct lack of information with my console outputs, I opted to move away from basic console outputs in the “DatabaseContoller” class to using log4j to add more in-depth logging to improve this. Developed and maintained by Apache, log4j is well known for its ease of use, performance, and plugin support to further extend functionality (The Apache Software Foundation, 2024). Leveraging Log4j’s capabilities, outputs contained more information than standard outputs, allowing any SQL errors and time stamps to be logged.

While this modification is not immediately noticeable to the end user, it significantly streamlined the debugging and development processes. Implementing Log4j enabled more efficient troubleshooting. Below, is an example showcasing the utilisation of Log4j when attempting to create a new table with errors in the SQL statement.

|  |
| --- |
|  |
|  |
|  |

To reduce clutter and confusion when using the system an admin panel is used. Here users with the required access level will be able to modify the email information for sending emails along with viewing, editing and removing backed-up data and help documentation.

## Testing plan and results

The testing plan and results section of this report aims to provide a comprehensive overview of the approach taken to validate the functionality, performance, and reliability of the developed system. This section presents detailed results of testing through the use of a testing table. A testing table serves as a centralised repository for all testing-related information. Each row in the table represents a specific task, scenario, expected result, actual result, status and any additional notes. If needed, further details on individual tests and any changes made due to the result will be provided; along with this, a GitHub issue for any bugs encountered will be created.

### Testing table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Scenario** | **Expected Result** | **Actual Result** | **Status** | **Additional Notes** |
| 1 | Attempted to log in to the system – correct login information | The system will verify the login details and allow the user in. | The user is successfully logged in. | Pass. | n/a |
| 2 | Logging in with the incorrect details. | The system should block the login and display a message. | The user is unable to log in and an error message is shown. | Pass. | n/a |
| 3 | Selecting a payroll record. | When a payroll record is selected, a detailed record should be shown. | As expected, a detailed record is provided when selecting a month. | Pass. | n/a |
| 4 | Selecting an employee record. | The text fields are populated with the correct data. | Text fields display the data of the selected employee. | Pass. | n/a |
| 5 | Updating an employee record. | Checks if the email is being updated if so, also update the login information. Successfully updates the record. | All information is correctly updates, including the login email if required. | Pass. | n/a |
| 6 | Adding an employee. | The information from the text fields is used to create a new employee record. | An employee record with a unique ID is created from the information provided. | Pass. | If the user attempts to add a user that already exists a message is shown and this action is not performed. |
| 7 | Pressing the Clear button. | All text fields are cleared. | Test fields are set back to their default state – empty with hint text. | Pass. | The access level combo box does not have hint text after clearing it – this is a constraint of JavaFX. |
| 8 | Deleting an employee record. | The employee record, along with all relevant data is removed from the database. | Some employee data is correctly kept for data retention, the user itself is deleted. | Pass. | n/a |
| 9 | Deleting currently logged in employee. | User is unable to delete a record for the currently logged in user. | The current user record is not deleted and a simple error is shown. | Pass. | n/a |
| 10 | Updating personal details. | Once the edit button has been pressed, all text fields are editable and once pressed again the database is updated. | Text fields are unlocked ready for modification, some are not as they do not make sense for the user alone to change – this is intended. The database is correctly updated. | Pass. | n/a |
| 11 | Adding help. | A popup is shown that allows the user to add support information. Once the ADD button is pressed this is added to the database and shown in the table. | Help information is correctly stored in the database along with the information of the user who added it. | Pass. | n/a |
| 12 | Adding help – no data. | There is no attempt to add the data to the database. | An error message is shown. | Pass. | n/a |
| 13 | Adding help – full data. | The information is added to the database and displays on the help page. | The help information is correctly added and displayed. | Pass. | n/a |
| 14 | Deleting help. | The record is removed from the database. | The record is removed from the database. | Pass. | n/a |
| 15 | Updating help. | The correct help record is updated with the changes. | When updating a help record, modifications are made to the database and the added by is updated to the currently logged in user. | Pass. | n/a |
| 16 | Updating the email information. | The new email and password override the default data and is then used to send email when required. | New email data is correctly stored and emails can now be sent. | Pass. | Correct email details are required for the system to send any emails. Without there are errors are shown in the console but no crashes occur. |
| 17 | Updating the schedule for an employee. | The user is able to set a start time and end time. | Each week’s data is updated individually and correctly, calculations for the planned hours, pay and pay deductions are automatic and updated. | Pass | n/a |
| 18 |  |  |  |  |  |
| 19 |  |  |  |  |  |
| 20 |  |  |  |  |  |
| 21 |  |  |  |  |  |
| 22 |  |  |  |  |  |
| 23 |  |  |  |  |  |
| 24 |  |  |  |  |  |

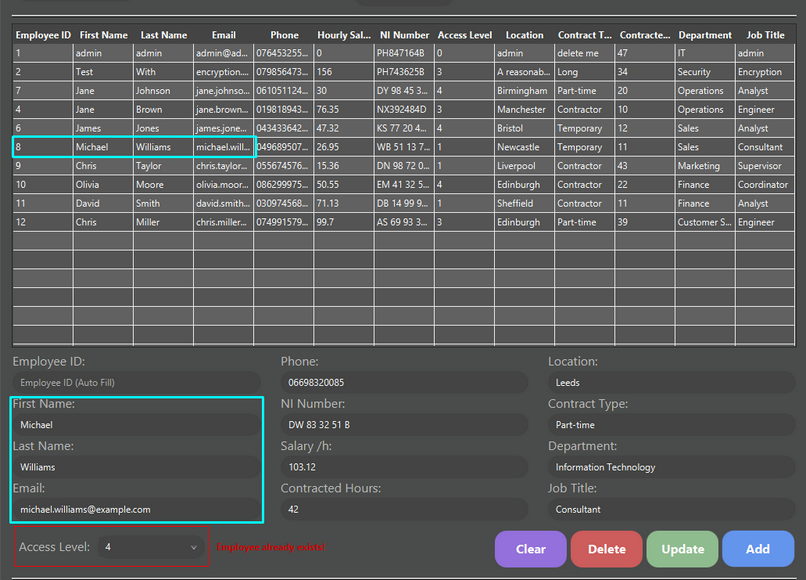
#### Test 2 – Logging in with incorrect details

When the user attempts to log in with the incorrect or no username and password simple error messages and text field highlights are shown to indicate there is an error.

|  |  |
| --- | --- |
|  |  |

#### Test 6 – Adding an employee

In the scenario when a user attempts to delete themselves from the database an error is shown.



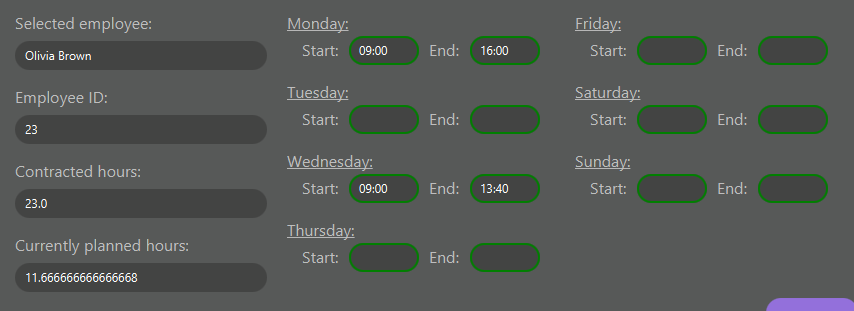
#### Test 15 – Updating help.

When an update is performed on the selected help record, any changes made are correctly stored in the database and the added by is updated to the currently logged in user.

|  |  |
| --- | --- |
| Before | After |
|  |  |

#### Test 17 – Updating the schedule for an employee

The user is able to enter a start time and end time for the selected employee, when the update method is called the database is updated and all relevant calculations are automatic.



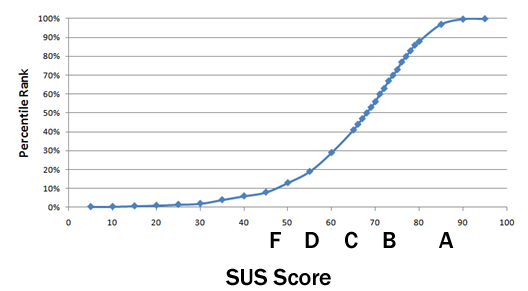
### System usability

To evaluate the system’s usability, the System Usability Scale (SUS) was employed. Originally developed by John Brooke, the SUS provides valuable insights into the user experience of the payroll application. A select group of peers participated in the assessment, responding to ten questions on a scale of one to five, with one being strongly disagree and five being strongly agree. The questionnaire was completed using Google Forms for easy participation and data collection.

The SUS offers a structured approach to evaluating the user’s experience and identifying areas to improve. The questions used were sourced from Digital.gov’s Usability Starter Kit (Digital.gov, 2024). Below are the ten statements from the questionnaire and, the original document can be found in the report’s appendix under the title The System Usability Scale (SUS).

|  |
| --- |
| 1. I think I would like to use this tool frequently.  2. I found the tool unnecessarily complex.  3. I thought the tool was easy to use.  4. I think that I would need the support of a technical person to be able to use this system.  5. I found the various functions in this tool worked well together.  6. I thought there was too much inconsistency in this tool.  7. I would imagine that most people would learn to use this tool very quickly.  8. I found the tool very difficult to use.  9. I felt very confident using the tool.  10. I needed to learn a lot of things before I could get going with this tool. |

The results will be important to understand the applications usability, identifying strengths and weaknesses. These insights will be used to further improve future iterations. As documented in 2011 by Doctor Jeff Sauro the average SUS score is 68, below is a graph showing the percentile rank and their associated SUS score and letter grades (Sauro, 2011).



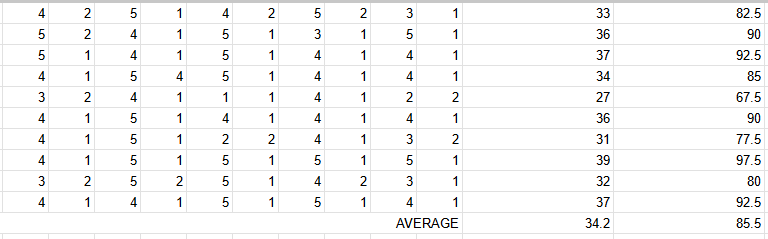
Each user was provided with a simple user guide that briefly covers each function of the application before being asked to take part in the test. Below is a copy of the user guide provided.

|  |
| --- |
| **User Guide**  **1. Logging In**   * Upon launching the application, you will be prompted to log in with your username and password. * Enter your credentials and click the "Login" button to access the system.   **2. Navigating the Dashboard and Navigation bar**   * Once logged in, you will be directed to the dashboard where you can access various features of the application. * The dashboard provides quick links to key functions such as managing employee records, viewing payslips, and accessing your person profile. * On the left-hand side there is a navigation bar, this is used to transition between the different pages available. What you see here will vary depending on your access level, if you cannot see something and believe you should please reach out to your manager.   **3. Viewing your Payslips**   * To view your unique payslip, navigate to the "Payslips" section in your profile. * Select the desired month to download the payslip for that period. * Save the payslip where you would like and open it with your favourite PDF viewer. * Enter the password (the last four characters of your national insurance number) to view the payslip.   **4. Viewing Payroll Data**   * To view payroll data, navigate to the “Payroll” page using the navigation bar. * Here you can view general data for each month on record. * Select a month to view specific data.   **5. Viewing and updating the schedule**   * Open the schedule page using the calendar icon in the system tray. * Select the week in the top right. * Find yourself or an employee using the search function. * With the correct permissions, select an employee and modify the text fields highlighted in green to change the start and end time. * Press update to submit these changes.   **6. Managing Employee Records**   * In the "Employees" section, with permission you can add, edit, or delete employee records. * Click on the "Add Employee" button to create a new employee record. * Select an existing employee to edit their details or delete the record if necessary.   **7. Accessing Help Resources**   * For assistance or guidance on using the application, navigate to the "Help" section. * Here you can search for helpful documentation, FAQs, and other details your colleagues have written. * To add new documentation, press the add button in the bottom right and fill in the form.   **8. Changing system theme**   * The system currently ships with two theme presets – light and dark mode. * Change between these by pressing the button on the bottom left of the application.   **9. Navigating the Admin panel**   * With the correct permissions you can see a shield icon in the nav bar – this is the admin panel. * Within this you’re able to view, edit and delete backed-up date, email server information and help information.   **10. Logging Out**   * To log out of the application, simply exit the application by right clicking the icon in the system tray and select “Exit Application”. * Logging out ensures the security of your account and prevents unauthorized access to your data but, remember to restart the application or no automatic processes will complete. |

Due to time constraints and the difficulty to share the application in its current stage, only ten participants were available. As seen results are reasonably equal across the board with no user having particularly struggled to use the application in its intended fashion. After completing the required calculations (Design, 2018), an end result was 85.5, which gives the application an A.

Testers commented on how they found the UI easy to navigate, with its clear layout and straightforward access to different pages. The consistency between pages was another positive mentioned however, some users attempted to full screen the application which is not fully supported; some pages will adapt whereas others do not. Error messages were praised for providing clear feedback and helping guide their interactions.

To gain a deeper understanding of this applications SUS score a more diverse and larger test group is required however, this suggests that the base application is currently easy to navigate and is user friendly.



A second Heuristic Evaluation Workbook had been completed. Both workbooks provide a systematic approach to assessing the systems usability and can be found in the appendix of this report.

Workbook 1 highlights the need for better user feedback regarding their current page, recommending the addition of titles and highlighting the active page for clarity. In workbook 2, improvements have been made, such as clear titles on each page, further enhancements could be made by adding highlighting to the navigation bar.

Improving upon the visuals of the system and accessibility, workbook 1 suggested the addition of a dark mode and clear error prevention and messages, which have been implemented. Workbook 2 expands upon these recommendations by suggesting error pop-ups in addition to the error messages to make them more apparent to the user.

Both workbooks emphasize the importance of keyboard shortcuts for increased user efficiency. Access control has been implemented between workbook 1 and 2, addressing the recommendation for personalised content.

Workbook 1 offered valuable insight into the application’s usability, which were incorporated into subsequent development efforts. Workbook 2 inspires further improvements that, along with the tester’s feedback, will guide additional improvements to create a more intuitive and efficient payroll system.

# Reflection

Looking back at my original objectives, I successfully researched existing payroll systems taking inspiration from these, deciding to create a prototype using Oracle APEX and the full application in Java and JavaFX. Before this project I had not used the Java programming language or the JavaFX framework, this meant I had to actively learn whilst designing and developing my application. I took this challenge on head first and am now comfortable whilst using both of these.

The current iteration of the payroll system can successfully manage the payroll process, allowing for scheduling followed by automatic calculations of pension and tax along with generating payslips which are send to each employee. Due to the system following an event-driven model and often relying on loops to iterate through records before performing an action, with a large enough database its plausible the system will begin to slow down. To improve upon this, I can look into multithreading in java to ensure the concurrent execution of two or more actions, this would allow the continual use of the application even if thousands of records exist. It may also be in the projects best interest to move to a more accessible system, such as a website which would not require any additional installations.

Testing was completed whilst developing to ensure that as many flaws or vulnerabilities were detected and mitigated as early as possible. This process was effective as it ensured each element was tested and fixed immediately but, ultimately led to less testing being recorded, in the future I will ensure constant logging of tests. Automated testing could be used to further strengthen and streamline my testing and quality assurance process.

## Aims

The first aim set out for this project was to develop a payroll system that enhances the employee’s experience. The final application successfully enhances an employee’s experience through its automatic personalised payslips and convenient access to personal and payroll information. Considering my System Usability results, the current system is reasonably user friendly however, there is room for improvement with the user interface and features that will directly impact the experience. By conducting further surveys and feedback sessions I could identify pain points and areas that require improvements.

A major commitment throughout my project was to always think of data protection and privacy. The project adheres to regulations such as GDPR and has does not collect any uniquely identifiable data that is not required to protect user’s privacy. Furthermore, encryption, hashing, and restricting user privileges though access control levels all contribute to keeping user data secure. As data protection and privacy regulations evolve additional security measures may be required to continue adherence. Additionally, extra security steps such as multi-factor authentication and regular security audits would further strengthen the system.

Whilst designing and developing an application its useful to consider how it will scale in the future to allow enhancements and modifications as needed. The module structure of my application allows for new features without extensive refactoring. However, there are areas where scalability can be further optimised. By conducting comprehensive assessments, I can identify potential limitations of the current system architecture. One possible improvement is to explore a web-app and use systems such as Docker and Kubernetes to containerise the system. By containerising deployment and management will be easier whilst also allowing for horizontal scaling.

## Planning

The integration of GitHub, a Gantt chart, and Tello as project management tools contributed to completion of the project within the set timeframe. Employing a hybrid approach by combining elements of Waterfall and Agile provided a balance between structure and flexibility throughout development.

GitHub served as the centralised repository to store the project and, allow detailed tracking of progress, bugs, and features awaiting implementation using their issues feature. Having documented progress made a detailed overview of the project can be seen.

Initially, a Gantt chart was created which set estimated timeframes for each component of the system, providing visual representation of the project’s timeline and key milestones. Gantt charts are widely used when following the waterfall method, allowing a visual breakdown of tasks and their subtasks. As development progressed, I updated the percentage complete for each task however, it became apparent that some tasks required more time than anticipated; the hybrid approach taken allowed for this did not cause major disruptions as I was able to quickly re-establish a new goal and work towards that.

Tello emerged as a valuable tool for creating and organising lists. The Trello board for the project had three lists, “To Do”, “Doing”, and “Done”. This organisation aligned with agile’s focus on breaking down projects into manageable tasks and provided a clear overview of completed and outstanding items. Trello’s board layout also mirrors the Kanban board which is a common tool used in agile managed projects (Rehkopf, 2024).

By breaking down the project into smaller tasks using these tools I maintained a structured yet adaptable project timeline, which helped me achieve the goals of the project. Despite encountering occasional delays, reasonable adjustments were made to reduce or mitigate the impact. As a result, the project was successfully completed within the desired timeframe.

## Future improvements

Looking forward there are improvements I could make to the base code. To begin with, there is room to optimise my SQL queries. Throughout the application I utilise prepared statements however, if batch processing was implemented this would allow for multiple SQL statements to be executed and minimises database round-trips. It’s also worth considering using MERGE statements where possible as these efficiently handles both insertions and updates based on certain conditions and would reduce the amount of code in the “DatabaseController” class.

As has been completed with the “DatabaseController” class, applying the Single Responsibility Principle where possible throughout the program would improve modularity and reusability. Identifying more opportunities to apply design patterns such as Singleton and Factory would further promote maintainability and scalability within the system.

Implementing a logout feature would improve the user experience and security. Taking this further, a session manager that handles user sessions by automatically logging them out after a period of inactivity with again improve the security of the system.

As I had not used JavaFX before I felt limited in what I could do within the time frame, looking forward I could introduce visual methods to view data such as charts, graphs, and dashboard to provide users with an insight into the payroll data trends at a glance.

Finally, to better protect user data automated backups to a separate database should be implemented that allows regular back up of critical data. The ability to them restore and read from these backups would greatly improve user experience in the case of data corruption or a natural disaster.

## Conclusion

In summary, the project has successfully achieved its aims and objectives. It has developed a payroll system that enhances the employee experience by providing reducing the time and effort required for calculations and record-keeping, along with the automatic emails sent containing the payslip. Furthermore, the project has shown a commitment to data protection and privacy through its limited data collection and security measures. Finally, the project has been designed with future improvements in mind, although there are further optimisations in certain areas would enhance scalability and flexibility. Overall, the project has effectively addressed its aims and laid out a good foundation for continued development and, I am happy with the application and my personal growth during this project. Having not undertaken a project of this scale before, I can now see how important comprehensive planning is.

# References

BambooHR, 2023. *HR Management Software That Brings It All Together.* [Online]   
Available at: https://www.bamboohr.com/  
[Accessed 7 November 2023].

Design, H.-C., 2018. *Calculating SUS Score With Google Form.* [Online]   
Available at: https://www.youtube.com/watch?v=7OJjJevg1xk  
[Accessed 16 April 2024].

Digital.gov, 2024. *Usability Starter Kit.* [Online]   
Available at: https://digital.gov/resources/digitalgov-user-experience-resources/digitalgov-user-experience-program-usability-starter-kit/  
[Accessed 15 April 2024].

GeeksforGeeks, 2023. *Introduction of Database Normalization.* [Online]   
Available at: https://www.geeksforgeeks.org/introduction-of-database-normalization/  
[Accessed 22 December 2023].

GeeksforGeeks, 2023. *Why to Choose Java For Backend Development?.* [Online]   
Available at: https://www.geeksforgeeks.org/why-to-choose-java-for-backend-development/  
[Accessed 22 January 2024].

Godoy, M., 2022. *Connecting your APEX apps to Non-Oracle Databases with Oracle Autonomous Databases.* [Online]   
Available at: https://blogs.oracle.com/apex/post/connecting-your-apex-apps-to-non-oracle-databases-with-oracle-autonomous-database  
[Accessed 2 November 2023].

ico, 2023. *What are PECR?.* [Online]   
Available at: https://ico.org.uk/for-organisations/direct-marketing-and-privacy-and-electronic-communications/guide-to-pecr/what-are-pecr/  
[Accessed 7 November 2023].

Information Commissioner's Office, 2024. *Principle (e): Storage limitation.* [Online]   
Available at: https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/data-protection-principles/a-guide-to-the-data-protection-principles/the-principles/storage-limitation/  
[Accessed 10 April 2024].

Martin, R. C., 2009. In: *Clean Code: A Handbook of Agile Software Craftsmanship.* s.l.:Prentice Hall, pp. 139-140.

Mendix Technology, 2023. [Online]   
Available at: https://www.mendix.com/  
[Accessed 18 October 2023].

Mendix Technology, 2023. *Deployment.* [Online]   
Available at: https://docs.mendix.com/developerportal/deploy/  
[Accessed 18 October 2023].

Mendix Technology, 2023. *Mendix Adds Powerful New AI and Machine Learning Capabilities to its Market and Technology-Leading Enterprise Low-Code Platform.* [Online]   
Available at: https://www.mendix.com/press/mendix-adds-powerful-new-ai-and-machine-learning-capabilities-to-its-market-and-technology-leading-enterprise-low-code-platform/  
[Accessed 18 October 2023].

Mendix Technology, 2023. *System Requirements.* [Online]   
Available at: https://docs.mendix.com/refguide/system-requirements/#databases  
[Accessed 18 October 2023].

Microsoft , 2020. *Power Apps Guided Tour.* [Online]   
Available at: https://guidedtour.microsoft.com/guidedtour/guidedtour/power-platform/power-apps/2/4  
[Accessed 18 October 2023].

Oracle, 2023. *APEX + AI = Awesome.* [Online]   
Available at: https://apex.oracle.com/en/platform/ai/  
[Accessed 18 October 2023].

Oracle, 2023. *App Builder User's Guide.* [Online]   
Available at: https://docs.oracle.com/en/database/oracle/apex/22.1/htmdb/understanding-rest-data-sources.html#GUID-36EE20FE-C015-4CAC-9A3B-5D49223F7DE2  
[Accessed 2 November 2023].

Oracle, 2023. *Deployment.* [Online]   
Available at: https://apex.oracle.com/en/platform/deployment/  
[Accessed 2 Novembet 2023].

Oracle, 2023. *Low Code.* [Online]   
Available at: https://apex.oracle.com/en/platform/low-code/  
[Accessed 18 October 2023].

Oracle, 2023. *What is payroll software?.* [Online]   
Available at: https://www.oracle.com/uk/human-capital-management/payroll/what-is-payroll-software/  
[Accessed 18 October 2023].

OutSystems, 2023. *Default Platform Server and database configurations.* [Online]   
Available at: https://success.outsystems.com/documentation/11/setup\_and\_maintain\_your\_outsystems\_infrastructure/setting\_up\_outsystems/default\_platform\_server\_and\_database\_configurations/  
[Accessed 9 November 2023].

OutSystems, 2023. *Deploying OutSystems.* [Online]   
Available at: https://www.outsystems.com/evaluation-guide/deploying-outsystems/  
[Accessed 9 Novembe 2023].

OutSystems, 2023. *Discover the OutSystems low-code platform.* [Online]   
Available at: https://www.outsystems.com/low-code-platform/  
[Accessed 9 November 2023].

OutSystems, 2023. *OutSystems pricing.* [Online]   
Available at: https://www.outsystems.com/pricing-and-editions/  
[Accessed 9 November 2023].

Refsnes Data, 2024. *Java Regular Expressions.* [Online]   
Available at: https://www.w3schools.com/java/java\_regex.asp  
[Accessed 15 April 2024].

Sage Group, 2023. *Payroll.* [Online]   
Available at: https://www.sage.com/en-gb/payroll-software/  
[Accessed 9 November 2023].

Sauro, J., 2011. *Measuring Usability with the System Usability Scale (SUS).* [Online]   
Available at: https://measuringu.com/sus/  
[Accessed 16 April 2024].

The Apache Software Foundation, 2024. *Apache Log4j.* [Online]   
Available at: https://logging.apache.org/log4j/2.x/  
[Accessed 16 April 2024].

The Apache Software Foundation, 2024. *What is Maven?.* [Online]   
Available at: https://maven.apache.org/what-is-maven.html  
[Accessed 23 January 2024].

UK Government, 2023. *Data protection.* [Online]   
Available at: https://www.gov.uk/data-protection  
[Accessed 7 November 2023].

valto, 2023. *What is the best database for PowerApps?.* [Online]   
Available at: https://valto.co.uk/microsoft-powerapps/databases/  
[Accessed 18 October 2023].

Wave Financial Inc., 2023. *Manage your money like a boss.* [Online]   
Available at: https://www.waveapps.com/  
[Accessed 9 November 2023].

# Appendix

## Risk Mitigation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| R.No | Risk Type | Risk Priority | | | | Mitigation Actions |
| Impact[[1]](#footnote-1) | Likelihood[[2]](#footnote-2) | | Score[[3]](#footnote-3) |
| Technical | | | | | | |
| 1 | Oracle APEX shutting down temporally or having to roll-back data resulting in data loss or corruption. | 3 | 1 | | 3 | Due to the impact of this risk being quite high as it could result in work and time being lost, I will make ensure that automatic back-ups are available and, also create my own backups of my application. |
| 2 | Data breach | 4 | 1 | | 4 | Oracle APEX automatically encrypts all data stored, I will include data encryption in my Java application and, to prevent unauthorised access I will make use of access control so users can only access what they should.  To prevent access to my base application during development I will use a strong password and an encrypted password manager to store it. |
| 3 | Software bugs and errors | 3 | 3 | | 9 | Bugs and errors are guaranteed to happen, I will reduce the chances by planning every step out in advance.  I will also employ thorough testing and quality assurance processes. |
| Commercial | | | | | | |
| 1 | Budget overrun | 2 | | 1 | 3 | Going over budget is unlikely as I will be using the always free tier of Oracle APEX for the prototype, Java, JavaFX, and Oracle database are also all free.  If at any point I feel the need to spend money to complete the project I will assess the situation and if needed contact my supervisor for support. |
| 2 | Market changes | 2 | | 1 | 2 | I will stay up-to date with other payroll software trends and regulations relevant to my application. I’m unable to predict any changes however, there will likely be advance notice so I will be able to adapt as needed. |
| Project | | | | | | |
| 1 | Scope creep | 4 | 3 | | 12 | Scope creep can sometimes be difficult to notice therefor, I will define the project scope clearly. Any changes to the scope will be documented and evaluated for their impact on the project schedule. |
| 2 | Resource Shortages | 3 | 2 | | 6 | As APEX is a new system to me, I will need to learn the system before I can effectively complete the project. I will do this through, YouTube tutorials, documentation and Oracles own tutorials. These sources are unlikely to go anywhere, mitigating any concerns.  Java and JavaFX are also new to me, I will learn this via the same techniques as above. |
| 3 | Timeline delays | 4 | 2 | | 8 | A delay can add additional stress and reduce the quality of the end product. To reduce the chances of delays I will use a Gantt chart to monitor and manage my projects timeline.  If there are unexpected delays I will reassess and adapt as needed. |
| Environment | | | | | | |
| 1 | Infrastructure Failures and natural disasters. | 4 | 1 | | 4 | APEX automatically creates back-ups and has an in-depth disaster recovery plan.  For my Java based application there is less protections in place, I will ensure my application’s code is backed up on GitHub. |
| Ethics | | | | | | |
| 1 | Data privacy violations | 4 | 1 | | 4 | I will create the application to comply with data protection regulations. Access to sensitive data will also be limited to authorised personnel only.  During development of the application, I will not be using any real-world data. |

## Heuristic Evaluation Workbook - 1

**Nielsen Norman Group**

[**https://www.nngroup.com/articles/ten-usability-heuristics/**](https://www.nngroup.com/articles/ten-usability-heuristics/)

**1**

# Visibility of System Status

**Issues**

**Recommendations**

**Issues**

**Recommendations**

Might not be clear what page user is on.

**The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time.**

Add title to each page and highly the current page in the navigation menu.

Does the design clearly communicate its state?

Is feedback presented quickly after user actions?

**2**

**Match Between System and the Real World**

Application uses mostly plain English.

**The design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon. Follow real-world conventions, making information appear in a natural and logical order.**

Will user be familiar with the terminology used in the design?

Do the design’s controls follow real-world conventions?

**Issues**

**Recommendations**

**Issues**

**Recommendations**

## User Control and Freedom

**Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action without having to go through an extended process.**

In the top right there will be a drop-down menu where the user will be able to log out – accessible on every page.

Being able to roll-back to previous versions of the database.

No visible way to log out of user account.

No undo or redo function shown.

Does the design allow users to go back a step in the process?

Are exit links easily discoverable?

Can users easily cancel an action? Is *Undo* and *Redo* supported?

**4**

Design is consistent through-out the application.

Add an optional “dark-mode”.

### Consistency and Standards

**Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform and industry conventions.**

Does the design follow industry conventions?

Are visual treatments used consistently throughout the design?

#### Error Prevention

**Issues**

**Recommendations**

**Issues**

**Recommendations**

**Good error messages are important, but the best designs carefully prevent problems from occurring in the first place. Either eliminate error-prone conditions, or check for them and present users with a confirmation option before they commit to the action.**

Add pop-ups if data inputted is incorrect or if an error occurs.

Clearly display what can be inputted and prevent submission if incorrect.

No sign of error messages.

No input constraints or validation shown.

Does the design prevent slips by using helpful constraints?

Does the design warn users before they perform risky actions?

**6**

**Recognition Rather**

**Than Recall**

**Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g. field labels or menu items) should be visible or easily retrievable when needed.**

Further labelling of items will help inform the user.

Minimalistic design will help from overwhelming the users.

Application only shows what is needed for that page.

Does the design keep important information visible, so that users do not have to memorize it?

Does the design offer help in-context?

Additional options for UI such as, dark-mode will improve user experience.

Design of each page only shows the essentials and keeps to a minimalistic design.

**Flexibility and Efficiency of Use**

**Issues**

**Recommendations**

**Issues**

**Recommendations**

**Shortcuts — hidden from novice users — may speed up the interaction for the expert user such that the design can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.**

No keyboard shortcuts or touch gestures.

Application will use access control to limit what each user can see and modify as needed.

Does the design provide accelerators like keyboard shortcuts and touch gestures?

Is content and functionality personalized or customized for individual users?

**8**

**Aesthetic and Minimalist Design**

Additional options for UI such as, dark-mode will improve user experience.

Design of each page only shows the essentials and keeps to a minimalistic design.

**Interfaces should not contain information that is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility.**

Is the visual design and content focused on the essentials?

Have all distracting, unnecessary elements been removed?

**Issues**

**Recommendations**

**Issues**

**Recommendations**

**Help Users Recognize, Diagnose, and Recover from Errors**

No error messages shown.

Help section not clear who or where the request is sent too.

Provide errors messages as needed, ensuring they are easy to understand.

Help section should include text explaining the process.

**Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.**

Does the design use traditional error message visuals, like bold, red text?

Does the design offer a solution that solves the error immediately?

##### 10 Help and Documentation

**It’s best if the system doesn’t need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.**

Write simple documentation and include in the help page.

No documentation available.

Is help documentation easy to search?

Is help provided in context right at the moment when the user requires it?

## Heuristic Evaluation Workbook - 2

**Nielsen Norman Group**

**https://www.nngroup.com/articles/ten-usability-heuristics/**

**1**

# Visibility of System Status

**Issues**

**Recommendations**

**Issues**

**Recommendations**

**The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time.**

A title has been added to provide more information to the user however, by highlighting the currently selected page in the navigation bar it will be clear at a glance what page is currently open.

It is not shown in the navigation bar what page the user is currently on.

Does the design clearly communicate its state?

Is feedback presented quickly after user actions?

**2**

**Match Between System and the Real World**

By incorporating different languages it will make the application more accessible to a wider range of potential users.

The application is only available in English, it uses clear and concise language as to now overcomplicate tasks.

**The design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon. Follow real-world conventions, making information appear in a natural and logical order.**

Will user be familiar with the terminology used in the design?

Do the design’s controls follow real-world conventions?

**3**

**Issues**

**Recommendations**

**Issues**

**Recommendations**

## User Control and Freedom

**Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action without having to go through an extended process.**

Implement a log out feature to return the user to the login page without fully restarted the application

By creating confirmation pop-ups before any action is executed it will ensure user’s are sure they want to make these changes.

Although a log out method had been considered it was not a priority in development and remains unavailable.

Once a change has been made there is limited ways to reverse this.

Does the design allow users to go back a step in the process?

Are exit links easily discoverable?

Can users easily cancel an action? Is *Undo* and *Redo* supported?

**4**

No recommendations.

Design is consistent through-out the application.

The additional dark mode is consistent and adds to the accessibility of the system.

### Consistency and Standards

**Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform and industry conventions.**

Does the design follow industry conventions?

Are visual treatments used consistently throughout the design?

#### Error Prevention

**Issues**

**Recommendations**

**Issues**

**Recommendations**

**Good error messages are important, but the best designs carefully prevent problems from occurring in the first place. Either eliminate error-prone conditions, or check for them and present users with a confirmation option before they commit to the action.**

Additional pop-ups will ensure user’s see the error.

There are currently error messages that are shown when an input fails validation.

Does the design prevent slips by using helpful constraints?

Does the design warn users before they perform risky actions?

**6**

**Recognition Rather**

**Than Recall**

No recommendations.

Information on each page is kept to a minimum, what user’s see changes depending on their access level.

**Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g. field labels or menu items) should be visible or easily retrievable when needed.**

Does the design keep important information visible, so that users do not have to memorize it?

Does the design offer help in-context?

**Flexibility and Efficiency of Use**

**Issues**

**Recommendations**

**Issues**

**Recommendations**

**Shortcuts — hidden from novice users — may speed up the interaction for the expert user such that the design can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.**

By including keyboard shortcuts, it can increase a user’s efficiency and also improves accessibility.

There are no keyboard shortcuts or touch gestures available.

Visible content is personalised based on a user’s access level.

Does the design provide accelerators like keyboard shortcuts and touch gestures?

Is content and functionality personalized or customized for individual users?

**8**

**Aesthetic and Minimalist Design**

No recommendations.

The design of each page only shows the essentials and keeps to a minimalistic design.

**Interfaces should not contain information that is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility.**

Is the visual design and content focused on the essentials?

Have all distracting, unnecessary elements been removed?

**Help Users Recognize, Diagnose, and Recover from Errors**

**Issues**

**Recommendations**

**Issues**

**Recommendations**

Adding more information to the system user guide could further help users.

Error messages are clear and any text field causing a problem are editable.

The help page is accessible by all and can show further information to assist users.

**Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.**

Does the design use traditional error message visuals, like bold, red text?

Does the design offer a solution that solves the error immediately?

No recommendations.

A user guide along with a basic privacy policy has been written.

##### 10 Help and Documentation

**It’s best if the system doesn’t need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.**

Is help documentation easy to search?

Is help provided in context right at the moment when the user requires it?

## The System Usability Scale (SUS)

This is a standard questionnaire that measures the overall usability of a system. Please select the answer that best expresses how you feel about each statement after using the website today.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Strongly Disagree | Somewhat Disagree | Neutral | Somewhat Agree | Strongly Agree |
| 1. I think I would like to use this tool frequently. | □ | □ | □ | □ | □ |
| 1. I found the tool unnecessarily complex. | □ | □ | □ | □ | □ |
| 1. I thought the tool was easy to use. | □ | □ | □ | □ | □ |
| 1. I think that I would need the support of a technical person to be able to use this system. | □ | □ | □ | □ | □ |
| 1. I found the various functions in this tool worked well together. | □ | □ | □ | □ | □ |
| 1. I thought there was too much inconsistency in this tool. | □ | □ | □ | □ | □ |
| 1. I would imagine that most people would learn to use this tool very quickly. | □ | □ | □ | □ | □ |
| 1. I found the tool very difficult to use. | □ | □ | □ | □ | □ |
| 1. I felt very confident using the tool. | □ | □ | □ | □ | □ |
| 1. I needed to learn a lot of things before I could get going with this tool. | □ | □ | □ | □ | □ |

How likely are you to recommend this website to others? (please circle your answer)

Not at all likely 0 1 2 3 4 5 6 7 8 9 10 Extremely likely

## Ethics Form

**SECTION A: Project Definition**

**FOR UNDERGRADUATE & TAUGHT POSTGRADUATE ONLY**

**Complete the following table with full and relevant information relating to your research.**

|  |  |
| --- | --- |
| Student Name | Kyle Chaplin |
| Student Number | 30017080 |
| Student E-mail Address (please use University e-mail) | 30017080@students.southwales.ac.uk |
| Name of Principal Project Supervisor | Shiny Verghese |
| Project Title | Payroll system with Oracle Database |
| Briefly describe the project, being sure to identify any aspects that are relevant to the Ethical Evaluation in Section B.  NOTE: A project determined to be High Risk will need to include additional information in Section B to fully-specify the risks and mitigations. | The objectives of this project is to develop a Payroll application system using Oracle APEX. Data will be securely stored using the Oracle database. Due to the sensitivity of the data that is involved in such a payroll system a comprehensive research methodology will be applied to fully understand GDPR and other data protection acts/laws. The functionality of the system will include automation of payroll calculations, including tax, retirement savings, and other benefits offered for employees by concerned organisations. It will also feature an automated payslip generation that will be distributed via email. |
| Please add an explanation of your study in plain English, with particular focus on any parts of your study which involve human participants. No more than 100 words. This is to help the Faculty Research Ethics Committee (FREC) to understand the project. | The project does not involve any study with human participants. |

**SECTION B: Ethical Evaluation**

**FOR UNDERGRADUATE & TAUGHT POSTGRADUATE ONLY**

Consider the following points to determine the level of ethical risk your research presents:

1. Involves those who are considered vulnerable such as:

* Children under 16.
* Adults with learning difficulties.

Unless in an accredited setting, accompanied by a carer or professional with a duty of care.

1. Involves those who are considered highly vulnerable such as:

* Adults or children with diagnosed mental illness/terminal illness/dementia/in a residential care home.
* Adults or children in emergency situations.
* Adults or children with limited capacity to consent

1. Involves those who are “dependent” on others (such as teacher or lecturer to student). Unless in an accredited setting associated with normal working conditions or routines and within normal operating hours, such as a cultural institution, pre-school, school, or youth club where the research is carried out as part of professional practice such as curriculum development.
2. Requires full NHS ethical approval via the Integrated Research Application System.
3. Requires a Human Tissue Act license.
4. Involves “covert” procedures as in covert observation studies.
5. Involves anything considered “sensitive”. For example, does not carry a risk of those involved disclosing information which compromises the research (e.g., illegal activities; activities where moral opinion may differ, potential professional misconduct – work errors).
6. Induces significant psychological stress or anxiety, or produce humiliation or cause more than fleeting harm / negative consequences beyond the risks encountered in the normal life of the participants (and where the potential for fleeting “harm” is clearly detailed in the participant information sheet). If in doubt regarding definition of the above terminology please contact the research governance office.
7. Involves administration of drugs, placebos or other substances (such as food substances or vitamins) as part of this study.
8. Involves invasive procedures (not limited to blood sampling, collection of biological samples, or passing current through a participant’s body, etc.).
9. Offers any financial inducements to participate in the study.
10. Intends to recruit serving prisoners or serving young offenders via Her Majesty’s Prison & Probation Service.

For your course, there may be specific requirements in **addition** to these, depending on the nature of the subject and how your project is assessed. You must also complete those requirements.

If **none** of the 12 points above apply, then the research can be considered **Low Risk**, *unless your course identifies additional criteria relevant to your subject that would render it High Risk*. This Section is then signed off by yourself and your supervisor, and held on file for review by FREC.

If **any** of the 12 points applies, then the research is considered **High Risk** and students must bring the matter to the attention of their research supervisor immediately. **Research cannot then commence until mitigations for the risk are agreed by FREC**. Seek advice from your Supervisor, who can help you identify mitigations of the risk or redesign as a Low Risk project.

**All students must complete the section below, in collaboration with their supervisor.**

Please strike through the statement that **does not** apply.

1. An ethics review has been completed, and the project has been identified as Low Risk.
2. ~~An ethics review has been completed, and a High Risk was identified. I agree to explain how they may be mitigated below, and agree to abide by any conditions identified at this stage, by my Project Supervisor, the School or the Faculty. I understand that High Risk projects can only proceed with approval from the Faculty Research Ethics Committee.~~

|  |
| --- |
| Issues: (Include as much information as possible to help FREC members to understand the issues. Extend onto additional pages as necessary.) |
| Proposed mitigations: (Include as much information as possible to help FREC members to understand the mitigations. Extend onto additional pages as necessary.) |
| Student’s Signature:  Date: 01/11/2023 |
| **Supervisor’s statement:** I have ensured due diligence and accountable decision making by the student. I have sought appropriate advice where required to support my judgment in this.  Supervisor’s Signature: A black and white image of a check mark  Description automatically generated  Date: 2/11/2023 |
| **Any false or mis-represented information contributing to this Ethical Evaluation, including attempting to pass off a High Risk project as a Low Risk project, is subject to the Student Misconduct Regulations and may also have legal repercussions.** |

Both signatures are **required** for all projects, both Low Risk and High Risk.

# Changes from milestone 1

#### Milestone 1 feedback:

In response to the feedback received during milestone 1, enhancements have been made to refine the project objectives [Page 6], improve the transitions between paragraphs, and ensure a more academic tone throughout my report.

Since the hand-in of milestone 1, the project has undergone significant changes and enhancements. The changes collectively represent a substantial leap forward from milestone 1. Below is a summary of any changes or additions along with the page number where relevant.

#### Database design:

The initial database [page 21] has undergone refinement, aligning it more closely with the requirements of the project [page 32]. This has resulted in a database that supports the functionality of the project and, scalability and data integrity.

#### Oracle APEX prototype and the move to Java:

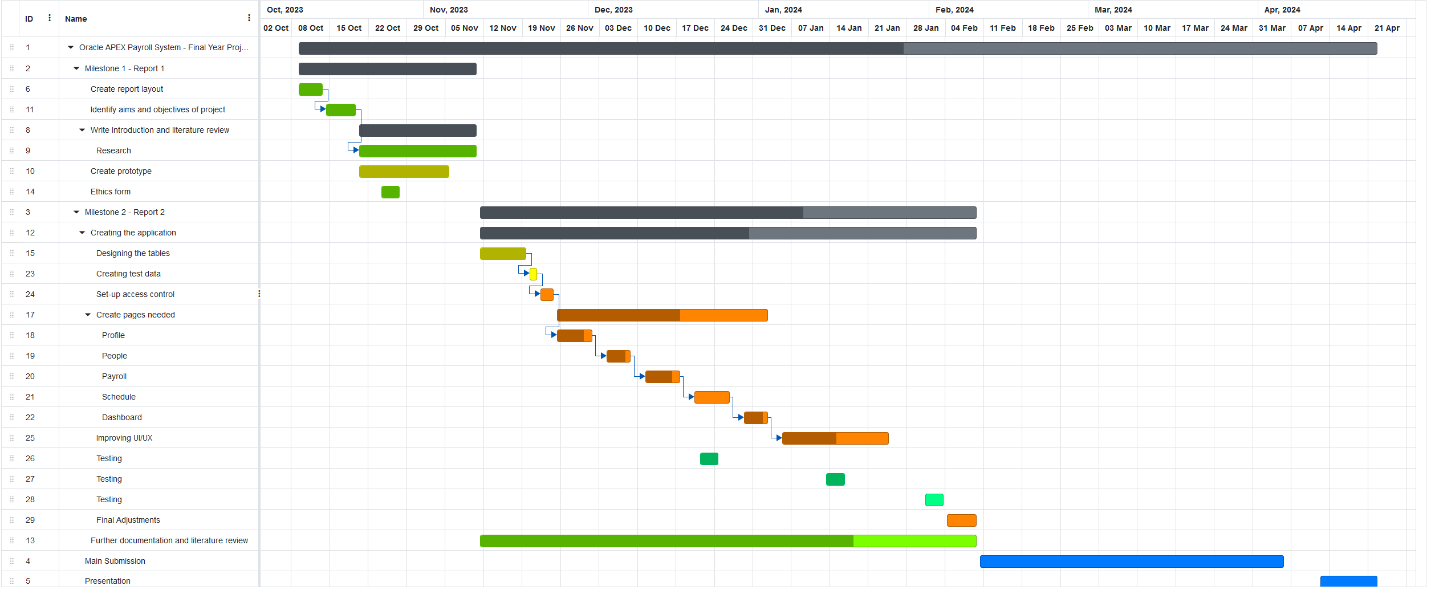
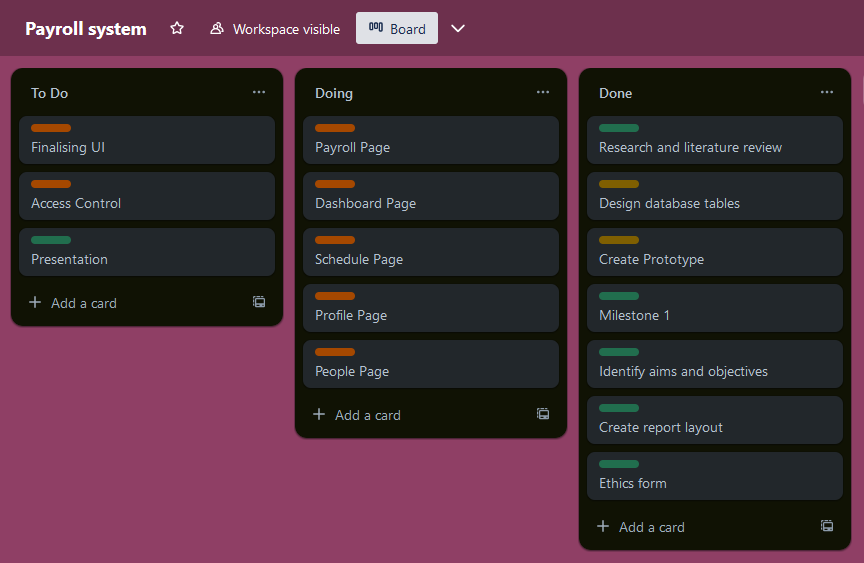
The Oracle APEX prototype served as a crucial offering to the system by providing a visual representation of the interface and functionality. The move to Java and JavaFX has been instrumental in providing greater control and flexibility over the development of the payroll system.

#### Feature implementation:

Currently, the dashboard, payroll, employee, profile, and help page all have their major features implemented and working as intended [page 26]. A core feature planned for my system was the automatic emails and PDFs [page 34] – this is also present as of milestone 2. Looking back on plans, the time required to implement the system was underestimated however, I remain confident and pleased with the progress I have made thus far.

The tracking tools, including the Gantt chart, Trello, and GitHub issues, have played a vital role in monitoring and managing progress. Regrettably, the current status, as reflected in the Gantt chart, indicates a deviation from the set schedule. In response to this, a reassessment of priorities has prompted a modification to the original plan.

Recognising the pressing timeline constraints, the decision has been made to set aside the development of a fully functional and interactive calendar. Instead, the plan now prioritises the creation of a streamlined method for employers to assign shifts to their employees. This aligns more closely with the core objectives of the payroll system, ensuring that critical functions such as payroll calculations are still possible. While the calendar feature remains outside the immediate scope of the payroll system, if circumstances allow, the calendar will be revisited for reassessment.



Along with the addition of many features, improvements have been made to the user experience through designing and implementing a light and dark mode toggle [page 38]. The current themes are not final and refinements are anticipated as development continues. The user interface modification, including the themes, contributes to an elevated user experience. As the project advances, attention will be directed to optimising the user interface.

#### Risk mitigation:

The risk mitigation table has undergone revisions in response to changes in the project [Appendix – Page 56]. As the project develops, objectives and priorities may adapt, introducing new elements that require an assessment of potential risks. The current changes relate to the move to Java as the programming language, a review was conducted, no major risks were identified, and the table remains mostly the same. A continued commitment to reassess any potential risks as progress is made towards milestone 3.

#### Looking forward – milestone 3

Key steps that need to be taken before milestone 3 encompass finalising core functionalities, integrating a robust validation process, and encryption to improve the database and application security further. Thorough testing and documentation will continue in this stage. Additionally, a user guide and fundamental privacy policy will be written. Looking ahead, I am enthusiastic to refine my application, knowing that each successful step brings the project closer to its final form.

1. Scale 1 – 5, 5 being highest. [↑](#footnote-ref-1)
2. Scale 1 – 5, 5 being highest. [↑](#footnote-ref-2)
3. Impact × Likelihood [↑](#footnote-ref-3)