Taiye, Jebutu [19067126]

Swift Solutions | VErsion – 1.4 19/07/2023

Project Proposal

Automated Payroll System

Connor, Day [18011054]

Connor Day [18011054]

Alexander, Wall [19067702]

David, Harvey [20000604]

Table of Contents

[1.0 Managing Software Development Projects 2](#_Toc140666503)

[1.1 A Comparison of Common Development Strategies 2](#_Toc140666504)

[1.2 A choice of Development strategy 3](#_Toc140666505)

[2.0 Project Outline 4](#_Toc140666506)

[2.1 Problem Statement & Project Background 4](#_Toc140666507)

[2.2 Client Requirements 4](#_Toc140666508)

[2.3 Business Case - Project Goals & Benefits 5](#_Toc140666509)

[2.4 General Aims & Objectives 6](#_Toc140666510)

[3.0 Project Management and Development Strategy 6](#_Toc140666511)

[3.1 Preferred Approach to Project Management and Development 6](#_Toc140666512)

[3.2 Provisional Project Gantt Chart and Task Schedule 7](#_Toc140666513)

[3.3 Project Risks 8](#_Toc140666514)

[4.0 Team Structure & Setup 9](#_Toc140666515)

[4.1 Team Members & Team Roles 9](#_Toc140666516)

[4.2 Software and Project Management Team Skills Matrix 10](#_Toc140666517)

[5.0 Development, Testing & Deployment 10](#_Toc140666518)

[5.1 Target Platform 10](#_Toc140666519)

[5.2 Development and Testing Platform 11](#_Toc140666520)

[5.3 Project Collaboration and Sharing 11](#_Toc140666521)

[5.4 Link to Online Repository 11](#_Toc140666522)

[System Requirements 11](#_Toc140666523)

[6.1 Existing System Use-Cases 12](#_Toc140666524)

[6.1.1 System Requirements 13](#_Toc140666525)

[6.2 Project’s Functional Requirements 13](#_Toc140666526)

[6.3 Project’s Non-Functional Requirements 15](#_Toc140666527)

# Managing Software Development Projects

In this section we will get a brief introduction into 3 software development strategies detailing origin, similarities between strategies, overall approach of the strategy, typical implementation and advantages and disadvantages. This will be followed by a choice of development strategy in which the justification will be given for said strategy.

## 1.1 A Comparison of Common Development Strategies

**Waterfall:**

In 1970 Winston R Royce wrote an article titled “Managing the Development of Large Software Systems” in which he defined a list of steps that he believed to be “fundamentally sound” for developing Large Software Systems. The overall approach consists of: First, getting System Requirements and Software Requirements, next Analysis of the requirements, then Program Design, Coding (Implementation), Testing, and finally, Maintenance (Operations).

Advantages:

* Great for documentation
* Requirements are clear and they remain the same throughout a project.
* Predictability
* Low reliance on client / Minimum Client Intervention.
* Inflexibility.

Disadvantages:

* Inflexibility.
* Less communication with clients.
* Test fails could cause massive delays.
* Documentation is time-consuming.

**Agile:**

In 1995 Ken Schwaber and Jeff Sutherland presented Scrum to the public for the first time. Then, in 2001, 17 Software Developers including Ken and Jeff developed and released the Agile Manifesto defining 4 Key Values and 12 Principles. Implementation of Agile involves a great deal of collaboration with the client, and minimising work that needs to be done.

Advantages:

* Collaboration with users leads to higher satisfaction.
* Increase in flexibility allows for shorter development times, thus reducing costs.
* Much better collaboration within working teams.

Disadvantages:

* Less documentation.
* Heavy dependency on stakeholders collaborating.
* Higher demand on time (Sprint meetings, backlog refinement, etc.).
* Easy to get sidetracked due to no pre-defined process.

**Extreme Programming:**

XP was developed by Kent Beck in 1996, he released 12 practices much like the agile manifesto. Since XP is an agile framework, the overall approach is very similar to that of agile. The main differences include: Typically, shorter sprints, the workload can be changed mid-sprint, work is done in order of prioritization which is defined by the customer and XP defines its own engineering practices (e.g. Test-driven development).

Advantages:

* Cost cuts for software development.
* Higher quality-of-life for developers
* Close contact with customers.
* Code tested thoroughly.

Disadvantages:

* No emphasis on code quality (Code only has to pass tests).
* Can be difficult if developers are not in the same location.
* Can be additional work.

**Similarities:**

Waterfall – Agile

There aren’t many similarities between waterfall and Agile, but the steps defined by W Royce are still used, they’re just done for each bit of work rather than for all the work.

Waterfall – XP

Since XP is very similar to Agile, there also aren’t many similarities between the two.

Agile – XP

XP is an Agile framework, so it’s based on Agile. Everything defined in the Agile manifesto is also done in XP.

## 1.2 A choice of Development strategy

We’ve chosen to use an Agile development strategy. A large reason for this is because Agile allows us to produce usable prototypes on a regular basis. Despite the requirements being pre-determined and not likely to change at all (Which shouldn’t matter because Agile allows us to work on each requirement on an incremental basis), we still think Agile is the way forward because the massive time restrictions mean under a Waterfall methodology it would be unlikely we would get to the end of a single iteration. The reason for not going with XP is again because of time constraints. We don’t have enough time to use Test Driven Development. Instead, the Agile framework we will be using is Scrum, we already know how it works and can get on with work quickly and confidently. Also, we don’t know what work is going to be involved in the development at this stage and how long it might take which suits Scrum perfectly.

# Project Outline

This section provides a comprehensive overview and outline of the project, diving straight into the problem statement and background, the client requirements, project goals and benefits then covering the general aims and objectives. This will allow us to explain the project, showing understanding of what is required for this project.

## 2.1 Problem Statement & Project Background

The client is an organization which now has a significant enough number of employees that the current system is inefficient enough to make investing in an automated payroll system worth the cost. They have multiple office locations and have employees that work in the field.

The current system requires attendance to be tracked manually by using clock in/out cards and timesheets. HR then needs to manually enter, collate, and record attendance. This system is required for HR to calculate payroll for each employee.

This system has numerous issues which have become a larger problem as the organization has grown which include:

* Human error can cause clock in/out cards or timesheets to be inaccurate, which then require manual intervention to resolve.
* It is possible for employees to manipulate their attendance by proxy attendance and forging information on timesheets.
* Employees have tedious tasks to complete to record attendance and calculate payroll.
* Lack of transparency to due individual tasks and systems for documenting and analyzing the information.

By introducing an automated payroll system, employees (especially HR) will have more time to commit to other operational tasks and projects and will need to spend less time resolving issues related to the system due to the improved accuracy, thus increasing overall business efficiency. As well as improving the day-to-day experience for employees who no longer must complete these tedious tasks required by a manual system, which could improve employee satisfaction.

## 2.2 Client Requirements

Based on the information provided, the client requirements are detailed below using the MoSCoW prioritization method.

**Must Have Requirements:**

1. Each employee can login to the app with their own unique ID and password.
2. When an employee logs in into the system, an image is immediately captured, and GPS location recorded.
3. When an employee logs out of the system, an image is immediately captured, and GPS location recorded.
4. While the employee is logged into the system, their GPS location is recorded every 5 minutes.
5. The system automatically sends all user images captured and GPS locations recorded to the admin.

**Should Have Requirements:**

1. Admins can create and manage employee profiles for each employee by entering all relevant personal information.
2. Admins can check salary details and breakdown of the salary of each employee.
3. The system accurately records the working days and hours of employees based on what time they have logged in and out of the system.

**Could Have Requirements:**

1. The system automatically calculates the payroll for employees based on working days/hours, salary, and leave.

## 2.3 Business Case - Project Goals & Benefits

Our first primary business goal is to streamline payroll processes and enhance efficiency. Optimizing efficiency is essential for top performance and developing this system will increase work output and staff time efficiency. By automating payroll tasks, such as data entry and calculations, this can minimize errors and risks associated with manual processing. With the integration of GPS tracking and image capture, the system will enable accurate tracking of employee attendance, work hours, and location. This ensures that each employee fulfils their allocated time and contributes to the success of the client's organization. By streamlining payroll processes, our goal is to improve efficiency, accuracy, and minimize potential risks. This allows the client to focus on strategic initiatives and maximize their overall performance.

Our second primary business goal is to strengthen data security and compliance. As we develop the automated payroll system, it is crucial to implement robust security measures and ensure compliance with regulations such as GDPR. This aligns with our first goal of streamlining processes in the current landscape where companies are automating previously manual tasks. With the transition to digital platforms, securing the data we input becomes paramount. By prioritizing data security and compliance, we aim to protect sensitive information that was traditionally recorded on paper and ensure its integrity in the digital realm.

For project success, the client must provide system requirements, insights into the previous system, existing payroll data, and employee information. GDPR compliance challenges related to image and location tracking should be considered. Resources must be allocated for user training and support upon project completion.

The anticipated benefits and return on investment for the client in choosing this project include cost savings from eliminating manual management and data entry and less staff will be required, improved reporting and analytics for strategic planning, enhanced compliance, and security with GDPR and location tracking, and prevention of fraudulent entries into the system under false employee identities.

Potential impacts of the client not pursuing this project include increased administrative burden, higher risk of human errors and inconsistencies, data security risks, compliance challenges, and limited insights/reporting capabilities.

## 2.4 General Aims & Objectives

As a team we have decided to choose the following aims and objectives as they give the team a straightforward endpoint in which we can aim for. The purpose of these goals and objectives is to give us a clear direction and focus on our work. The goals are written with S.M.A.R.T in mind.

Goals:

1. Produce a software solution that is maintainable and scalable.
   1. Source code will contain user guides for the functions and modules written.
2. Produce releasable demos/prototypes at the end of each sprint.
   1. Main branch to have minor and major updates.
3. Produce a software solution that is compliant with industry standards.
   1. Follow python PEP 8 standards.
4. Software Solution to be accessible through the Web and not just on local networks.
   1. Incorporate a deployment method for the software solution.
5. Adhere to agile methods of working.
   1. Follow the SAFe Agile Framework

# Project Management and Development Strategy

*Provide* *a short introduction about the purpose of this section of the report (up to 50 words)*

This section of the report will describe the general management of the project regarding how we will approach the problem to develop a solution within the planned milestones in our Gantt chart. All while considering possible risks and how best to mitigate them.

## 3.1 Preferred Approach to Project Management and Development

After reviewing the common approaches used to manage and develop projects, especially software projects as a team, we have decided to manage and develop this project within an agile approach. The decision to stick an Agile methodology of working is because we as a team agreed with the core principles of the agile manifesto which discusses prioritising working software over documentation, which in the period given for this project seems the best approach to take with this project. As discussed in section 1.2 of this report, we will be using scrum which is a framework of the Agile methodology also. This has been decided due to the software developers within our team already having vast experience with this process which is much faster to train the other two members of the team on this rather than adopting an entirely new approach nobody is familiar with.

Reviewing our agile approach critically, we can see the benefits from using this approach with the first benefit being that it allows adaptability to change and adapt to added information and feedback, allowing continuous improvement, this is important as means we are flexible to any added information or feedback from our client whilst developing this project. Another benefit of this approach is using agile allows for improved collaboration and communication, enabling regular meetings with the developers and stakeholders closer together encouraging transparency and allowing the team to be aligned with the project goals and letting everyone be on the same page. The final main benefit of this approach after this review is also the fact that it allows for risk mitigation, as compared to other methodologies, this approach allows for continuous progress and maintaining the project at all times meaning any risks which appear can be captured within the following development cycles.

Now looking at the possible risks and issues associated with the agile approach to software project development and management, firstly is the risk of the scope creeping and expanding beyond the original scope's boundaries. As the requirements can change frequently due to the flexible nature of this approach, this can lead to the scope becoming further away from the original scope set. To mitigate this, we will establish a well-defined initial scope with the client and prioritise each requirement set on importance this combined with a change management process will enable the impact evaluation of any changes to our initial defined scope from expanding too far. Another risk associated with using this approach for our project is that communication involves dependency on your team and the stakeholders involved to be available for meetings. This is a risk which will be a problem for us that we have considered with our team working in different environments with different schedules, however we have agreed to manage this risk by setting weekly sprint meetings which everyone can agree to with also the added feature of the meeting chair taking minutes so if someone has to leave or cannot join, they do not miss any information discussed.

## 3.2 Provisional Project Gantt Chart and Task Schedule

In order to provide a solution for our client in time as a team we have come together to break down the tasks into what we call features. Each feature works towards implementing a requirement for our project. Internally within our team features will be broken down into stories, of which when completed, the feature can also be considered completed. E.g., the feature ‘Research Python Web Capabilities’ would be broken down into stories internally such as, *research Python database creation, research python web login system, research python web modules such as flask and Django*. The features that are called ‘Complete Section …’ are related to the completion of each section of this project proposal document. Some tasks are to be done at the same time because they do not conflict with each other and allow team members to work at the same time and not have to wait and do nothing.

## 3.3 Project Risks

The inclusion of image and location tracking introduces additional data security concerns and regulatory compliance risks. There is a risk of unauthorized access, data breaches, or misuse of sensitive employee information captured by the system. To manage this risk, implementation of robust security measures, including encryption protocols, access controls, and secure storage practices, will need to happen to safeguard employee data. Conducting regular security audits and assessments to identify vulnerabilities and address them promptly. Comply with relevant data protection regulations such as GDPR to ensure data security and privacy policies are met at all times.

Employees may resist the change from manual payroll processes to the automated system. They could be wary of the new tracking functionalities or just simply face challenges in adapting to accessing the new system, leading to low user acceptance and resistance to adopt it. To manage this risk, conducting comprehensive user training along with clear documentation (I.e., a manual) is required to be provided to ensure employees understand the benefits and functionalities of the new system. If the client allows for different users of the system to test prototype systems to provide feedback this will further enable, clear management of this risk.

# Team Structure & Setup

This section details our team's structure for this proposed project and how we have chosen to set it up. Our individual specialisms and skills are used to provide team roles and carefully determine each team member’s strengths and weaknesses to best meet possible success for this project being proposed.

## 4.1 Team Members & Team Roles

In Swift Solutionswe have 4 members from the following companies, Leonardo, Arrow and Exemplas which will be described below.

Introducing our team members and roles, we can start with Taiye Jebutu. Taiyes’ team title is Scrum Master and Software Engineer, the scrum master title is responsible for managing the exchange of information and facilitating/assisting the success of the Agile methodology we have chosen to use for this project. Alongside this title and responsibility, he is coming from a software engineering background which will involve writing source code and managing the prototyping process, this background of experience will enable him to develop the software to a professional standard.

The second member of our team is Alex Wall. Alex’s team title is Software Engineer, his responsibility is revolving around supporting Taiye with the software development of the project, whilst also taking responsibility for his own side of writing source code and researching solutions to adapt into the project to maximize efficiency. Alex coming from a background of software engineering will also allow him to use this experience to ensure the highest quality of software is developed for the client.

The third member of our team is David Harvey. David’s team title is Data Analyst, his responsibility includes designing the database and how it will interact with the software that we will develop for the automated payroll system whilst also documenting the source code which is written by Taiye and Alex allowing communication between the software development team and other teams within the project to be free flowing.

The fourth member of our team is Connor Day. Connor’s team title is IT support, his responsibility includes designing the user interface for the software application, ensuring the logical needs and requirements for the client are met through the software development whilst also having the additional responsibility of being the solutions functionality tester, allowing a high detail of logic adding to our software to ensure each solution is necessary, functional and required.

|  |  |  |
| --- | --- | --- |
| Name | Title | Responsibility |
| Taiye Jebutu | Scrum master, Software Engineer | * Organize and setup team working environment * Write Source Code * Prototyping leader |
| Alex Wall | Software Engineer | * Write Source Code * Solutions researcher |
| David Harvey | Data Analyst | * Database designer * Source Code documentation |
| Connor Day | IT Support | * User Interface designer * Solution Functionality tester |

## 4.2 Software and Project Management Team Skills Matrix

0 = no experience, 1 = Basic knowledge, 2 = intermediate, 3 = Very good, 4 = Expert

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Taiye | Alex | David | Connor |
| Technical Writing | 2 | 2 | 2 | 2 |
| Research | 2 | 3 | 2 | 2 |
| Graphic Design | 2 | 1 | 2 | 3 |
| Programming | 3 | 3 | 2 | 2 |
| Database Design | 1 | 1 | 3 | 1 |
| Web Design | 1 | 1 | 1 | 2 |
| Software Design | 2 | 3 | 2 | 2 |
| Problem Solving | 3 | 3 | 3 | 3 |
| Analytical Skills | 2 | 1 | 4 | 3 |
| Project Management | 1 | 1 | 2 | 1 |

Team activities and tasks will be assigned while taking into account the skills of each team member, to ensure that work done is to a high quality and team members are given an opportunity to improve their weaknesses.

# Development, Testing & Deployment

In this section we will briefly discuss the target platform (e.g., web) and delve into the specifics behind said platform. We’ll look at how we intend on developing and testing this application and the platforms involved in that process. And finally, the intended methods of collaboration used on the project.

## 5.1 Target Platform

Given the nature of our project we will be creating a web-based application, which will be exclusively accessible via a company intranet. Unlike the internet, which is a global network of computers that is accessible to anyone, an intranet is a private network that is only accessible to users within an organisation. An extranet also exists, this is a private network where external parties have access to certain parts of an intranet. This intranet will need to be hosted from somewhere, this could be from a cloud-based server but, given the size of the application will be relatively small we think a company network server intranet option would be more appropriate. Prior to accessing the web application, some form of identification will be required (username and password). This information will be stored in a MySQL database system along with personal data belonging to each user. Front-end development will be done using HTML while back-end development will be done in Python.

## 5.2 Development and Testing Platform

As a team we came together to discuss which development platform will be the best for us to use. We considered personal experience, ease of use and whether this development platform will enable us to create a solution for this project. In the end we chose to use PyCharm, as our IDE (integrated Development Environment) to support a solution made in python. PyCharm has an in-built functionality to create virtual environments to ensure that the development environment of all the team members match and removes the issue of solutions working on one computer and not another.

PyCharm also has Git implemented into the IDE which can also link to GitHub for well-built version control. As a team we are using GitHub to provide collaborative work on this project, which gives us another reason for using PyCharm.

In addition to the reasons above, the latest version of PyCharm provides the functionality of real-time pair programming on a project. Which means that, if need be, the same source file on the same branch can be worked on by more than 1 person while on a call to each other. This further improves our collaborative ability.

## 5.3 Project Collaboration and Sharing

For this project GitHub will be used to enable collaborative work. GitHub provides a fully comprehensive version control system when used in conjunction with Git. This is important because we are using SAFe (ScaledAgileFramework) agile which requires a comprehensive version control system. This means that multiple team members can work on different parts of the program without conflict and once their task is done and reviewed it can be merged into the main project for a seamless way of working.

## 5.4 Link to Online Repository

Below is the link to our online repository for the Automated Payroll System project we are undertaking.

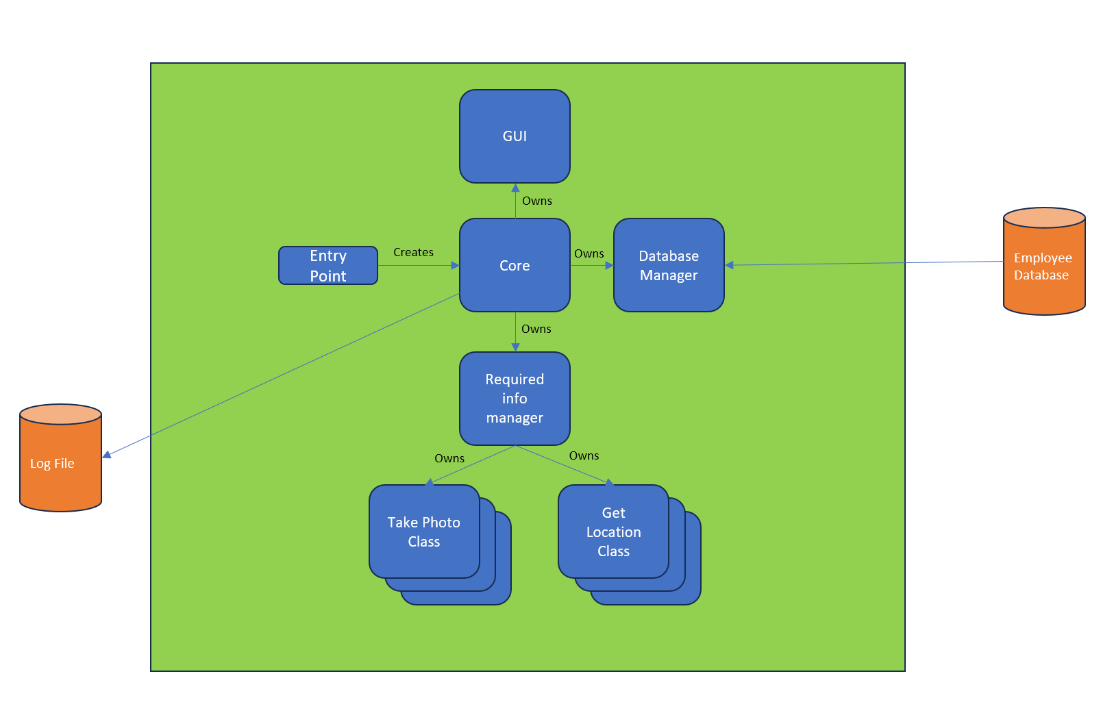
[GitHub - TaiyeJebutu/APS](https://github.com/TaiyeJebutu/APS)

# System Requirements

For this section, we will be discussing the existing system our client chooses to use to show our knowledge of the proposed project we have gained, we will then be leading into identifying functional and non-functional requirements for this project and how we then manage and prioritise these.

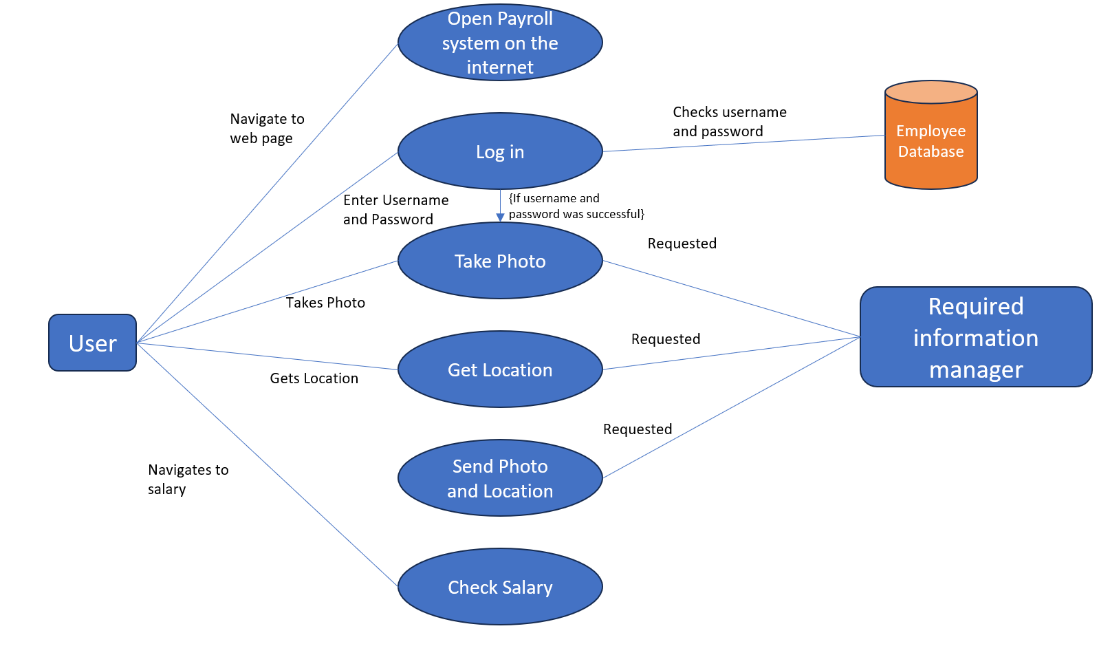
## 6.1 Existing System Use-Cases

**Basic original design created to help create use case diagrams:**

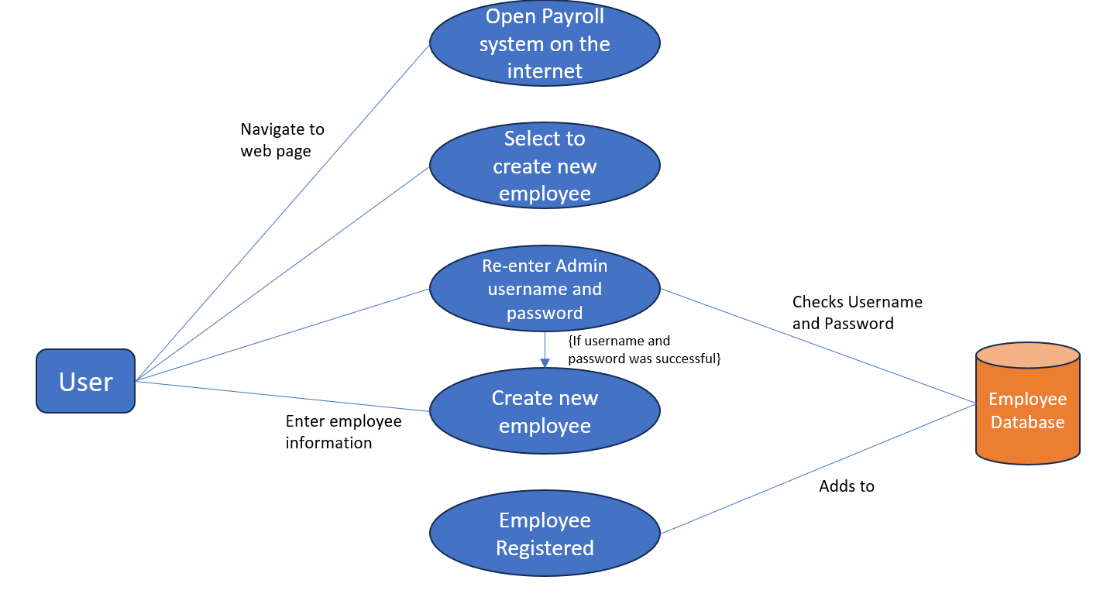


The following are use-cases for the client’s ideal system.

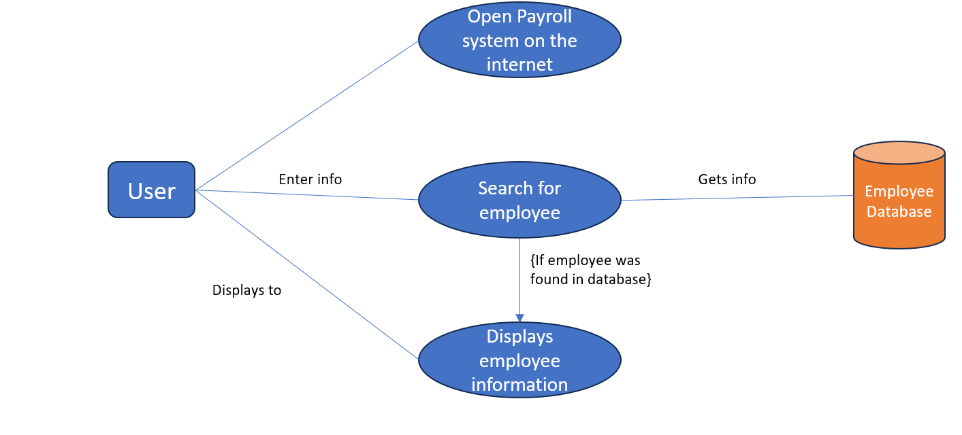
**Use Case 1 – User logs in and checks salary details:**



**Use Case 2 – Admin creates a new employee profile:**



**Use Case 3 – Admin checks an employee’s information:**



### 6.1.1 System Requirements

Functional requirements define what the system must do and how it should respond to certain behaviors. They should specify features and capabilities that the software we are developing must provide to meet client and user requirements.

Non-functional requirements describe how the system should perform rather than what it should do. They are the constraints imposed on the system and specify the quality attributes of the system, of which will have a significant impact on the quality and success of the software we develop

## 6.2 Project’s Functional Requirements

1. **The system must allow users to log in by entering their unique ID and password.**  
     
   Without this requirement, the system will not know which user in the system to attach the photo and location to.
2. **The system must capture a photo when a user logs in and out of the system.**  
     
   Ensures users cannot be logged in by proxy (by another employee) and unauthorized users without it being obvious to the admins (as the picture will be a different person).. This is important because the client wants to ensure that the information of each user is recorded accurately and only allows the correct users to access the system.
3. **The system must record the user’s GPS location when they log in and log out, and every 5 minutes whilst logged in.**  
     
   This is important because the client wants to know their users are working where they should be. This also provides a set of data of which anomalies can be spotted which could improve security e.g., a login from a country of which the client does not operate in or a user logging in and out of locations 5 hours away from each other, within 1 hour.
4. **All images and GPS locations must be sent to the admin.**  
     
   This allows the client to track who is signing into the system, at what time and at what location so they can ensure no identity theft is taking place. This will also be important for recording hours worked to calculate payroll for each employee.
5. **The system shall allow admins to create, retrieve, update, and delete employee profiles and information.**  
     
   Admins will be HR administrators or whoever the client deems to be administrators of the system. This includes employee personal details, payroll information, salary details and job roles. Which enables admins to maintain accurate employee records and ensures necessary information for payroll management is available. Without this requirement, the system cannot be maintained and kept for new users to be added to the system correctly.
6. **A unique ID and Password must be generated upon creation of a new user.**  
     
   An ID and password are generated which can be sent to the user, ensuring they can login for the first time.
7. **The system should establish a link between unique IDs (and their password) generated by associating them with the employee for whom they were generated.**  
     
   This ensures that when a user logs in with their unique ID and password, that the information captured is for each unique ID, is matched with the correct employee.
8. **The system shall allow users to view their own payroll information upon login.**  
     
   This is a needed requirement as it is the main feature of this system, allowing users to see what their pay slips are and other personal documents/information.

## 6.3 Project’s Non-Functional Requirements

**Performance:**

1. **The system should respond to requests within at least 2 seconds.**  
      
   This ensures a good user experience and that users will not lose attention and patience with the system (which would cause the system to be less effective due to lack of interest). Whilst also ensuring the task of logging in and out doesn’t become a chore for employees. Users should be able to perform their tasks more efficiently than in the previous system. This requirement considers that some users could be working in remote locations with very poor signal, but the system should still respond within an appropriate time once the request has been received.

**Usability:**

1. **Users (including admins) should be able to use all functions (that they have access to) after 1 hour of training. After training, there should be no more than 1 request for assistance per week on average.**  
      
   Using the system should be easier than the previous solution of recording attendance, to obtain engagement and enthusiasm for the app from the employees, it should make their job easier than the previous solution did.

**Security:**

1. **The system shall not allow users to log in if they do not have an account.**  
      
   This is important to prevent unauthorized access/identify theft, as well as preventing data without a user associated being input into the database.
2. **Employee data should only be accessible by admins.**  
      
   This is important to ensure the system meets legal requirements, as well as the data policies of the company.

**Availability and Reliability:**

1. **The system should be accessible whilst a user is connected to the company network.**  
      
   Whilst a user is connected to the company network, they must be able to access the system, regardless of the location of the user and the device they are using. Without this requirement, some users may not be able to login which will mean attendance will be inaccurate. It is also specified directly by the client that they want the system to be accessible from both the web and android devices.
2. **Providing that that company infrastructure and network are fully operational, the system should be available and accessible 99.95% of the time every month during business hours.**  
      
   This system requires high uptime during business hours, otherwise employees will not be able to login, resulting in inaccurate attendance and HR may not be able to calculate payroll in time.

**Scalability:**

1. **The system should be capable of managing twice the number of employees currently employed at the beginning of each year.**  
      
   This ensures as our client’s organization grows, the system is scalable with the growth of the business. Otherwise, the system would be rendered ineffective if it reached its full capacity.