**Timesheets System Requirements Specification**

Introduction

This project is intended to create a convenient, efficient, and secure way of storing data based on how the organisation’s employees are spending their time on projects. The current solution involves each employee having a basic spreadsheet that must be manually created each month. The spreadsheets are not secure because they can be accessed by any employee, and the spreadsheet is not backed up. Analysing this data is also very difficult and inefficient because the data is collected in different spreadsheets, and data is inputted in different ways by different users. The new timesheets system will create a web-based system that has user authentication and authorisation levels to increase security. The new system will also collect data as an organised and structured database, so the data can be quickly and easily queried to analyse the data and make the data more valuable.

Potential Issues

* The system may not be completed because the project has a low resource priority. This is an in-house project, which will create a system that is only used internally; this means that externally commissioned projects that generate the organisation revenue will have a higher priority.
* In the future, the system may require to be accessed outside of the organisation’s intranet
* The system must be able to scale, as the organisation grows. If the database is not designed effectively, system performance may be negatively impacted by the organisation growing.
* Scope creep may negatively impact the project’s timescales, required resources or quality

Assumptions

* It is assumed that the system will be used on desktops, rather than mobile devices, so the system will not be optimised for mobile use.
* It is assumed that the system will only be accessed by the organisation’s intranet, and does not need to be accessible remotely / publicly
* It is assumed that 2 user levels (employee and manager) levels are necessary
* It is assumed that the user will be accessing the system on a device with an internet connection

Prototypes

The Adobe XD prototype has been designed alongside the client, and has been signed off by the client, so it will form part of the service level agreement (SLA). The high-fidelity prototype highlights some functional and non-functional requirements

Non-functional Requirements

|  |  |  |
| --- | --- | --- |
| **Requirement reference** | **Name** | **Description** |
| req\_001 | Scalability | The system needs to be able to scale as the organisation grows. This will include more users actively using the system and more projects being worked on by the organisation, and more user time records being saved |
| req\_002 | Accessibility | The system needs to be accessible via a web browser by desktop machines that are on the organisation’s intranet |
| req\_003 | Reliability | The system needs to be reliable, with minimal downtime |
| req\_004 | Recoverability | The system, and it’s data, needs to be recoverability in the event of data loss |
| req\_005 | Security | The system needs to be secure. The data should have strict access rights, with only the minimum amount of access rights being given to users, based on how they need to use the system. Any sensitive data being stored should be encrypted. The system should account for common methods of attack like SQL injection and brute force attacks. |
| req\_006 | Usability | The system should be easily usable and intuitive to users with a range of ages, and technical ability. The system should be usable on a desktop, being viewed on desktop monitors |
| req\_007 | Maintainability | The system should be easily maintainable by the development team. The system should have clear abstraction, comments, and separation between units of functionality. |

Functional Requirements / User Stories

UserStoryRef-1: As a manager, I want to be able to register new users so I can create new employees for my department.

Acceptance Criteria:

* Verify if valid user details are entered into the registration form, a user is registered
* Verify if invalid user details are entered into the registration form, validation errors are displayed

UserStoryRef-2: As a manager, I want to be able to register new clients so I can add my new clients to my department.

Acceptance Criteria:

* Verify if valid department details are entered into the department creation form, a department is created
* Verify if invalid department details are entered into the department creation form, validation errors are displayed

UserStoryRef-3: As a manager, I want to be able to create a new project so that I can add projects to my department as we get them commissioned.

Acceptance Criteria:

* Verify if valid project details are entered into the project creation form, a project is created
* Verify if invalid project details are entered into the project creation form, validation errors are displayed

UserStoryRef-4: As a manager, I want to be able to create a new project stage so that my employees can be more precise when assigning their time to the project

Acceptance Criteria:

* Verify a project stage can be created with valid details
* Verify if the user attempts to create a projet stage with invalid details, validation errors are displayed
* Verify the project stage has been added to the relevant project

UserStoryRef-5: As a manager, I want to be able to delete a project, with its project stages, so that I can delete old / completed projects from the system

Acceptance Criteria:

* Verify a project can be soft deleted
* Verify that the project’s stages are soft deleted deleted
* Verify that the project’s time records are not deleted

UserStoryRef-6: As a manager, I want to be able to delete a user so that I can delete user accounts for employees who leave

Acceptance Criteria:

* Verify that the user can be soft deleted
* Verify that the user’s time records are not deleted

UserStoryRef-7: As a manager, I want to be able to view analytics on a project and project stage so that I can analyse workflow and accurately quote for projects

Acceptance Criteria:

* Verify that the user can view how much a project has cost
* Verify that the user can view how much a project stage has cost
* Verify that the user can see how long a project has taken
* Verify that the user can see how long a project stage has taken

UserStoryRef-8: As an employee, I want to be able to login to the system so that I can view and add to my timesheets

Acceptance Criteria:

* Verify the user can go to the login webpage and enter valid credentials to gain access to their account
* Verify if invalid credentials are entered, the user is appropriately notified

UserStoryRef-9: As an employee, I want to be able to logout so that I can keep my account secure and inaccessible

Acceptance Criteria:

* Verify the user can logout
* Verify the user cannot access any data from the database

UserStoryRef-10: As an employee, I want to be able view my time records for a specific date so that I can review how I spend my time

Acceptance Criteria:

* Verify the user can list all of their time records for the desired date
* Verify that soft deleted time records are not listed
* Verify that the user is notified if they select a date where there are no stored time records

UserStoryRef-11: As an employee, I want to be able add a new time record to a specific data, project and project stage

Acceptance Criteria:

* Verify the user can create a new time record for a selected date and project, but without a project stage
* Verify the user can create a new time record for a selected date, project and project stage
* Verify that the projects listed when creating a new time record, are the project assigned to clients that belong to the user’s department
* Verify that the listed project stages when creating a new time record are the project stages that belong to the selected project

Test Plan

*Notes: the scenarios in the test plan should all be executed in Google Chrome, Firefox, Safari and Edge. Equivalence partitions will be used in the test procedure to save time without reducing test plan quality / coverage.*

Scenario name: Employee Login

Scenario user stories: UserStoryRef-8, UserStoryRef-9

Scenario script:

|  |  |
| --- | --- |
| Step Description | Expected Outcome |
| Navigate to the login page | The login form will be shown |
| Enter invalid login credentials | Validation errors will be shown |
| Enter valid employee login details | The user will be logged in and redirected to the dashboard |
| Click the logout button | The user will be logged out and redirected to the login page |

Scenario name: Employee Time Records

Scenario user stories: UserStoryRef-10, UserStoryRef-11

Scenario script:

|  |  |
| --- | --- |
| Step Description | Expected Outcome |
| Navigate to the login page | The login form will be shown |
| Login as an employee | The employee’s dashboard will be shown |
| Select a date on the calendar where there are no time records for the user | The date will be highlighted and the user will be notified that there are no time records for that date |
| Select a date on the calendar where there are time records for the user | The date will be highlighted and the user’s time records for the selected date will be listed |
| Create a new time record with a selected project but no project stages or minutes | Validation errors will be shown |
| Create a new time record with a selected project and enter minutes but not select a project stage | The new time record will be added to the list |
| Create a new time record with a selected project and project stage | The new time record will be added to the list |

Scenario name: User management

Scenario user stories: UserStoryRef-1, UserStoryRef-6

Scenario script:

|  |  |
| --- | --- |
| Step Description | Expected Outcome |
| Login as a manager | The manager will be logged in and redirected to their dashboard |
| Click users in the top navigation bar | The existing users in the system will be listed |
| Click the new user button | The form will be displayed where the details for a new user can be entered |
| Enter invalid details for the user | Validation errors will be shown |
| Enter valid details for the new user | The new user will be created and added to the list of existing users |
| Press the delete button on the user in the list | The user will be marked as deleted in the database and the user will be removed from the list on the user interface |

Scenario name: Client Management

Scenario user stories: UserStoryRef-2

Scenario script:

|  |  |
| --- | --- |
| Step Description | Expected Outcome |
| Login as a manager | The manager will be logged in and redirected to their dashboard |
| Click clients in the top navigation bar | The existing clients in the system will be listed |
| Click the new client button | The form will be displayed where the details for a new client can be entered |
| Enter invalid details for the client | Validation errors will be shown |
| Enter valid details for the new client | The new client will be created and added to the list of existing clients |

Scenario name: Project Management

Scenario user stories: UserStoryRef-3, UserStoryRef-4, UserStoryRef-5, UserStoryRef-7

Scenario script:

|  |  |
| --- | --- |
| Step Description | Expected Outcome |
| Login as a manager | The manager will be logged in and redirected to their dashboard |
| Click projects in the top navigation bar | The existing projects in the system will be listed |
| Click the new project button | The form will be displayed where the details for a new project can be entered |
| Enter invalid details for the project | Validation errors will be shown |
| Enter valid details for the new project | The new project will be created and added to the list of existing projects |
| Click on the project’s view button | The project’s details are listed. This will include the project’s stages and the project’s analytics. |
| Enter invalid details into the new project stage form | Validation errors will be shown |
| Enter valid details into the new project stage form | The project stage will be created and added to the list of existing project stages for the project |
| Click the projects button on the top navigation bar | The projects will be listed |
| Click the delete button on a project | The user will be asked to confirm to delete the project |
| Confirm the project deletion | The project will be marked as deleted and removed from the list of existing projects |

Solution Deliverables

* System: a full system should be delivered which will include a frontend, backend (API) and database. The system must meet all requirements and tests specified in the project’s requirements specification and overall SLA.
* Technical artifacts: an entity relationship diagram, UML class diagram, wireframe / prototype, requirements specification and maintenance plan must be delivered
* User training: training for end users must be supplied. This may be in the form of in-person training, remote virtual training, pre-recorded videos or written documentation.