Functional Specification Standard

Introduction

The Reimbursinator software suite will serve as an application for submitting reimbursement requests (reports?). This application will allow an administrator to create a travel policy and then allow requesters to submit reports, guiding them through the policy.

1. Introduction

1.1. Summary

Create an easy web application for companies to keep track of reimbursement requests that meets their policy.

1.2. Requirements

- User Roles
    - Submitter - Creates and submits reimbursement requests.
    - Administrator - Configures policy, receives reimbursement requests and edits database.
- User Stories
    - Accounts -
    - Database -
    - Policy -
    - Reimbursement requests -
    - Reports -

1.3. Numbers

- The expected amount of clients: IN PROGRESS
- How often: couple times a day
- peak usage time: day time/working hours
- Capacity of the platform:

1.4. Existing System

The purpose of the Reimbursinator is to replace the email system previously used to record travel reimbursement expenses for employees in large scale companies/industrial organizations. The current system is tedious and outdated; long email threads become tortuous and many important details are lost in translation. Our application will provide an accessible user interface for employees to easily input data, and for administrators to access completed forms. Although users will not be allowed to edit a submitted form, they will be given viewing access to their history - similar to how one would dig through their sent folder in their inbox.

1.5. Terminology

- Data Fields: Sub-fields of a section, used for storing values such as amount of money spent.
- Section: Sub-field of a policy, areas for handling rules involving differing travel expenses
- Reimbursement Request: A request for reimbursement for work related travel-expenses
- Report: The full reimbursement request, the document that will be submitted at the end

1.6. References

Forthcoming

2. Functional Description

2.1 Use Cases

- Web Application:
  - User login
  - Create account
  - (Pending) Update account details
  - Create reimbursement request
  - Update reimbursement request
  - Save section of report (in progress)
  - Submit report
- Backend Service
  - Supports interaction through REST API calls
  - Create report API call
  - Delete report API call
  - Get specified report (by ID?)
  - Get list of reports (By user ID?)
  - Update report
  - Update section

2.2 User Community

- Startups and small businesses

2.3 Administration Functions

The following functions will be available to the administrator.

- Configure policy file to create sections and rules for the reimbursement policy of the company/organization.

- Use administrator account to log in to the back end and view/edit database content directly.

2. 4 Error Handling

Errors(front end):

1. Leave blanks -> error notification
2. Wrong answers (not related / out of range) -> error notification
3. Platform error (cannot proceed to next step) -> prompt to refresh the page?
4. ...

Errors(back end):

1. Nothing return from the server -> technician
2. Wait for too long to go to next step -> prompt to refresh
3. Cannot load history or saved state forms -> technician
4. ...

2.5 Security

The primary security attacks our application is susceptible to surrounds user accounts and data communication. Users enter sensitive information such as bank account and credit card details to indicate their preferred method of reimbursement, making our system attractive to hackers. In order to combat these vulnerabilities, we are going to use token authentication for user login. Additionally, we will enhance security by using SSL/HTTPS protocols for data communication.

2.6 Help

Our application is straightforward to use - all the user has to do is enter the corresponding data in the generated form. Additionally, we'll provide helpful error messages in the situation a user enters an invalid value in the blank.

2.7 Printing

N/A

2.8 Interfaces

- Web app Ui provided for users accessing the application
- Command line access to database required for administrative changes

2.10 Boundary Conditions

- Use of different policies in same environment.
- User without the ability to change password
- Duration of reimbursement history

2.11 Constraints

- Code should be released under an GNU Affero GPL v3-compatible license
- The backend should be able to be run on a Debian server
- Submitters should be able to access the web application from both mobile and desktop platforms

2.12 Platforms

- Debian server
- Phone access/web server/desktop

2.13 Internationalisation

We currently don't have any plans to extend this product to an international audience. For this reason, our application will be in English.

2.14 Performance

We will be supporting multiple users at once, and this ideally shouldn't interfere with our response and communication times.

2.15 Portability

Although we plan to develop our application for a Debian environment, it should be easily portable to other Linux-based platforms.

2.16 Expandability

Currently we don't have any expansion requirements.

2.17 Customisation

- Allowed to modify the policy to their need
- Manually update the Database

2.18 Support & Maintenance

This project is planned with the assumption that the target audience has sufficient technical knowledge to procure and manage the necessary server environment, install and configure the Reimbursinator application, and maintain the code base. There are no current plans for the development team to provide support or maintenance beyond the product delivery.

2.19 Configuration Management

The scope of this project can only afford one software version, so we will

provide such management accordingly.

2.20 Documentation

The following documents will be produced upon completion of this project:

- Functional Specifications (Requirements)
- Technical Specifications (Design Plan)
- Software Verification & Validation
- Risk Management Plan