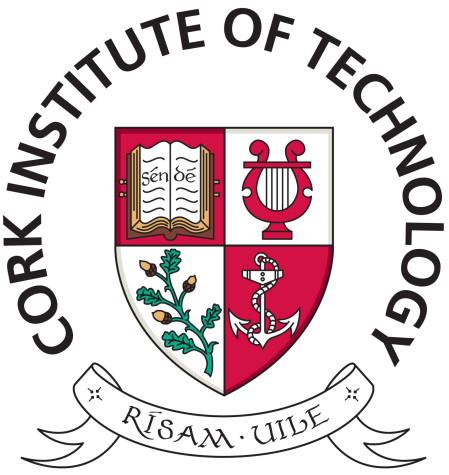


# Localisation Business Management System

by

Adam Lloyd

This thesis has been submitted in partial fulfillment for the degree of Bachelor of Science in Web Development

in the
Faculty of Engineering and Science
Department of Computer Science

December 2016

# Declaration of Authorship

I, Adam Lloyd, declare that this thesis titled, 'Localisation Business Management System' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for an undergraduate degree at Cork Institute of Technology.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at Cork Institute of Technology or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this project report is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

CORK INSTITUTE OF TECHNOLOGY

# *Abstract*

Faculty of Engineering and Science
Department of Computer Science

Bachelor of Science

by Adam Lloyd

To create a business management system which provides client, project and financial management along with a robust API to integrate this data into Intels existing localisation infrastructure and fulfils the requirements laid out by Intel Security.

# *Acknowledgements*

I would like to thank my family and friends who have supported me during my return to college and my supervisor Ruairi O'Reilly for advice and support throughout the year.

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **SQL** | **S**tructured **Q**uery **L**anguage |
| **HTTP** | **H**Hypertext **T**ransfer **P**rotocol |
| **SSL** | **S**ecure **S**ockets **L**ayer |
| **HTTPS** | **H**ypertext **T**ransfer **P**rotocol over **S**SL |
| **API** | **A**pplication **P**rogramming **I**nterface |
| **TMS** | **T**ranslation **M**anagement **S**ystem |
| **QA** | **Q**uality **A**ssurance |
| **CRM** | **C**ustomer **R**elationship **M**anagement |
| **ERP** | **E**nterprise **R**esource **P**lanning |
| **HR** | **H**uman **R**esources |
| **IT** | **I**nformation **T**echnology |
| **MVC** | **M**odel **V**iew **C**ontroller |
| **SaaS** | **S**oftware **a**s **a** **S**ervice |
| **CSS** | **C**ascading **S**tyle **S**heet |
| **REST** | **RE**presentational **S**tate **T**ransfer |
| **UI** | **U**ser **I**nterface |

*To Shiv*

# Chapter 1

# Introduction

For my final year project I have been asked by my former employer, Intel Security whom I worked with during work placement, to develop a business management system for the localisation department to replace their current system which is unsuitable for their current requirements.

## 1.1 Vision Statement

For software localisation teams who want an integrated financial and project management system. This will be a bespoke management system that will streamline the process of software localisation without the need to use multiple applications and be extensible without third party support.

## 1.2 Executive Summary

The localisation department of Intel Security is responsible for localising all of the content that Intel Security owns, this not only includes software but sales, marketing and support documents as well. To ensure that all of this content is correctly localized into all of the target languages, the expertise and resources of language service providers or translation vendors (referred to in this document as vendors) are leveraged by outsourcing much of the translation work to these specialised companies. These vendors work on a contract basis and managing them and the projects they are assigned to is an essential task.

Typically, a vendor will offer translation services for only a limited set of languages and may not be able to translate each of the languages (sometimes more than 50) that the

software supports. Similarly, each vendor will charge different rates per word for each language and different rates again for different types of words.

This presents a problem such that, for large localization projects, thousands of words must be translated into many languages all with varying rates often involving multiple vendors to meet the language requirements of the project. With existing systems, the management of vendors and the calculation of the payment due, is done manually resulting in time loss and increased risk of error.

Furthermore, with multiple projects running in tandem, the services of these vendors are limited and a record of which vendor is assigned to which project is needed to ensure accurate project planning can take place.

To facilitate this process, an application is required that will provide project managers with an interface to manage vendors, store the rates that they charge for their services, and calculate the payment that is owed once the contract is completed. It must also allow the translators themselves to submit the logs of their translation work once it has been completed and integrate with existing localisation tools for use by engineers.

Ultimately the goal of this project is to create a business management system which provides client, project and financial management along with a robust application programming interface (API) to integrate this data into Intels existing localisation infrastructure.

## 1.3 Structure Of this Document

Chapter 2 will explore the background of the project, business management and software localisation and the challenges they present.

Chapter 3 will bring chapter 1 and 2 together end begin to explore solutions to the previously mentioned problems driving the project.

Chapter 4 will expand on these solutions and detail the technology architecture of the application and methodologies to be used in the implementation.

Chapter 5 is a conclusion and will touch on future work that may stem form the completion of this project.

# Chapter 2

# Background

With the scope of this project being quite broad but also needing to fit quite a specialised role, it is hard to put it in a specific category. It is with this in mind that I have chosen to explore the background of the project under two main headings; software localisation and business management. There will also be a minor focus on software as a service bespoke versus generic software and the issue of outsourcing work.

## 2.1 Software Localisation

The localization of a software product involves separating all the content, usually text, within the application into separate files and having the application show the user the correct content based on the language they select to use with the application[1].

These language documents must be translated, and correctly localized, by expert companies known as language service providers or vendors for the purpose of this document. They will ensure that the content is in the correct language for the target locale and that it is in the correct context. They maintain a large database of previously translated words and sentences from previous projects which they reference to quickly translate commonly used words. Words that do not appear in their database will require more effort to translate and, as such will be costlier to translate.

### 2.1.1 Vendors as a resource

In large software development projects, multiple vendors may be contracted to localise the content as they will often not specialise in all of the languages required. This presents a need for these vendors to be managed as a resource as multiple projects may

be running in parallel. Were the vendors contracted to translate multiple projects at once, they would take longer to complete the work.

### 2.1.2 Localisation Work-flow

Traditionally management of vendors is done via manual emails and their work is completed using Translation Management Systems (TMS) such as GlobalSight [2] which will manage the work-flow from assigning the job to Quality assurance (QA) and acceptance by the client. While this project will not directly interact with these systems, it will need to integrate into a system that they are a part of and, as such, this is relevant to note as potential work-flow improvements involving other systems could be anticipated when developing this system even if they are not to be made yet.

## 2.2 Business Management System

Business management encompasses a range of functions including financial management, client management and project management all bundled together in one application. This project must provide all of the required functions in one place customized to fit the needs of the company exactly.

### 2.2.1 Client Management System

Client management, often called customer relationship management(CRM) is primarily a means of keeping track and in touch with customers to continue doing business with them and is most effectively employed by sales companies and the software of choice for many of them is Salesforce[3].

For the scope of this project, CRM will refer to the management of vendor accounts.

#### 2.2.1.1 State of the art - Salesforce

Salesforce, like many of the applications this document will look at is a large business intelligence system providing an all in one solution from financial management to analytics and enterprise level social networks [3]. It is the leading cloud based solution for business intelligence due, along with its long-standing reputation for excellence, to it being a cost-effective option for multinational companies as there is no need for on premise installation or hardware as all operations are done within a web browser.

However, with all of the robust features it provides, it requires 3rd party software to provide project management functionality which is required by the scope of this project.

### 2.2.2 Project Management

In many cases, managing projects is done via email or over the phone with the project manager contacting the client directly, sending on the details of the job manually and handling the inevitable back and forth privately. This often leads to costly delays due to confusion resulting from miscommunication and time management.

Project management refers to the creation and monitoring of a project from creation to completion. Within the scope of this system, projects can be created and a client can be assigned to them. Using the information from the saved client profile the client will be notified of the job that is assigned to them automatically via email. They will be able to view the details and status of the job and update the status as work progresses until completion.

The manner in which projects are being managed, at least in the software development space, has changed drastically since the emergence of the Agile methodology in 2001 [4]. This marked the departure from the conventional waterfall methodology which involved a specification document being produced then the development being completed based on that.

The waterfall system was flawed in that it did not react to changes in customer requirements as the project progressed and often resulted in incorrect and unwanted software being produced due to a lack of collaboration [5]. This is where agile comes in. With a manifesto touting short development cycles and close collaboration with all stakeholders throughout ensures useful software at the end of the project.

In order to manage short development cycles, increased collaboration and the constant changes which arise from this, increasingly complex management systems need to be created.

#### 2.2.2.1 State of the Art - VersionOne

One agile project management system of note is VersionOne, a dedicated agile project management system providing project time lines, backlogs, collaboration and analytics in a clear visual interface and drag and drop functionality for moving project tasks in and out of phases [6]. Like Salesforce, VersionOne is a cloud based offering making it

an excellent choice for multinational collaboration as all team members have access to the same information from anywhere in the world.

While Salesforce provided a number of services but lacked only one desired by the scope of this project, VersionOne specialises in only one. So, while lessons can be learned from researching the application, to use VersionOne for the project management component of this project would result in a massive number of unused features and ultimately a lack of integration with the other functions.

### 2.2.3 Financial Management

Financial management refers to the effective and efficient management of funds.

Financial management software is specifically designed to automate, assist and store financial information. It handles the storage, analysis, management and processing of a set of financial transactions, records and processes.

#### 2.2.3.1 State of the Art - SAP ERP

Enterprise Resource Planning (ERP) system. End to end solutions for financial management, CRM, logistics, human-resources (HR), and Sales [7].

At the time of writing, SAP maintains a 6 percent market share in the ERP space which is a full 2 percent higher than its nearest competitor which is worth 5.3 dollars billion in revenue.

SAP is primarily used by large companies and it is chosen due to its large range of modules which fulfil all ERP requirements. Furthermore, its popularity can be attributed to vendor lock in as it has been around since the early 80s in the form of its previous incarnation SAP R/3 [8]. IT departments and staff already familiar with SAP will likely continue to choose SAP as it reduces overheads on training and IT infrastructure.

SAP Aggregates data from all of these modules to provide important business insights at a high level and provides one solution to manage the entire business infrastructure of a company. It is an excellent choice for large multinational companies looking for an end to end solution to manage their entire business infrastructure form one solution. However, SAP is extremely expensive with full implementation of all modules taking years and costs reaching upwards of 100 million dollars once software, hardware and 3rd party consulting fees have been taken into account.

While its main strength is being an end to end solution for all business activities, to implement SAP as a financial component in a larger system would be overkill and the resultant consultation needed to integrate with the existing systems would be very costly.

Adding custom functionality to the system to account for the specific needs of a localisation department would also come at the cost of substantial time and money.

## 2.3 Bespoke vs Generic Software

Generic software refers to off the shelf applications purchased from a 3rd party while bespoke refers to Software created specifically for the purpose and to the requirements of the stakeholders. The reasons for choosing either are varied and always involve compromise in one aspect.

### 2.3.1 Cost

Cost is a factor that holds benefits for both types of software in different situations. For large, complex applications it is likely that the cost of developing a bespoke solution could be significantly higher than purchasing a generic solution. Development, testing and potentially outsourcing the development of the bespoke software will likely be far higher than simply purchasing an established generic solution. However, for smaller applications with few features development of a bespoke solution could prove cheaper without the need to compromise on functionality [9].

### 2.3.2 Functionality

To satisfy a wide range of customer requirements generic software will provide many features and be extensible with modules to create an all in one solution. This is an important selling point for these applications but the software will have been developed to fit general requirements not ones specific to a company's needs. This may result in having to compromise on some features and adjusting existing work-flows to fit the application. Conversely, a bespoke solution could be created with only the features required by the company and, by working together with the software developers, would fit their requirements exactly.

### 2.3.3 Maintenance

Any application will require maintenance regularly throughout it's lifespan and, while this is less of an issue with cloud based solutions, maintaining generic solutions could potentially be problematic [10].

Without the expertise or even access to the code, the company will likely be reliant on the generic software vendor to fix bugs, release updates and troubleshoot problems, often at a cost. With a bespoke solution, especially one developed by in house developers, these issues can quickly be resolved and potentially much cheaper.

### 2.3.4 Support

Popular generic solutions will likely have a wealth of support available from online forums to on to one support and support from the company that produced the software. When opting for a bespoke solution there will be no documentation, support available until the development team creates it further adding to the cost and time it takes to create the application.

### 2.3.5 Time

Complex applications can potentially take years to complete while generic software can simply be purchased, sometimes instantly online and operational in anywhere from minutes to weeks depending on the size of the organisation and a cloud based or on premise solution is chosen.

## 2.4 SaaS vs On Premise

One major benefit of a generic solution is that it can be provided as a cloud service. Software as a service (SaaS) is a model for providing software to a customer by removing the need for physical installations in their own offices but rather by providing a web portal with which they can log in to the software.

This model provides the customer with everything they need to use the software as well as all of the related infrastructure requirements such as IT support, updates and security. This is an attractive and affordable system but it is not always the right fit[11].

When using a shared cloud based solution, there is the potential for security and compliance breaches when dealing with sensitive data. The potential for network outages

and any other cloud provider problems also becomes an issue. Considering this, and the lack of direct control over the system and data, there is certainly an argument for an on premise solution despite the benefits of SaaS.

## 2.5 Outsourcing Projects

Of the types of project that this system will manage, the vast majority will be outsourced contracts which presents its own challenges.

Outsourcing is the contracting out of a service from one company and there a number of reasons a company would outsource an element of its business, primarily cost[12].

For instance, large software development companies may be great at producing software but there are many other processes that go into making the company run. They require accounting, HR, legal, security and a range of other services to support their software development and in many cases these roles may be outsourced. A company which specialises in software development can often find itself employing accountants, security guards and legal staff, fields which the company may have little expertise or staff for.

In cases like this outsourcing certain services is the most cost effective plan of action [12]. Instead of hiring a handful of staff to provide a service to a company can pay another company to provide that service, usually a company that specialises in that service and only that service.

### 2.5.1 Outsourcing Localisation

To bring this into perspective using localisation as an example, if a company supports only a few languages it may be wise to hire translators for those languages. If a business needs grow in the future, and begin to support more languages it would have to hire in additional translators.

Localisation service providers such as SimulTrans support 110 languages and are a dedicated translation entity [13]. They maintain a database of previously translated content which, after many projects will contain a substantial amount of content pre-translated which results in faster turnaround times and reduces cost and the increased quality inherent in a service from a specialised company.

Using such a company ensures an extremely high quality service with all of the expertise required to provide that service without hiring on any staff or training up existing staff to fulfil that role.

### 2.5.2 The Downside of Outsourcing

Managing the outsourced projects, the clients and ultimately paying them for their services [14] adds complexity to the work-flow. With in-house staff, they would simply be added to the payroll become a fixed cost. When outsourcing projects, the time worked would need to be accounted for accurately and paid for based on the time and effort put in. So, where there was a simple payroll there are now many other details and manage your finances and projects more accurately.

# Chapter 3

# Problem - Localisation Business Management System

## 3.1 Problem Definition

The core problem is to refine and automate the work-flow of the software localisation department of Intel Security by providing a modern advances web application to fulfil their requirements.

The problems with the existing system can be separated into two sections, the technical issues and the work-flow issues caused by a poorly optimised system.

### 3.1.1 Technical Issues with the Existing system

The current financial management tool has been used the Intel Security localisation team for over 10 years. There are numerous problems with it stemming from its development on top of an old bulletin board system as a temporary solution.

- It has been poorly planned as a result of being a temporary solution and this has resulted in features being tacked on with little thought for maintainability.
- The code is extremely insecure with plain text passwords being stored and transferred in many places.
- The database schema is poorly optimized and has many redundant tables
- The Technologies used are not consistent with other internal software

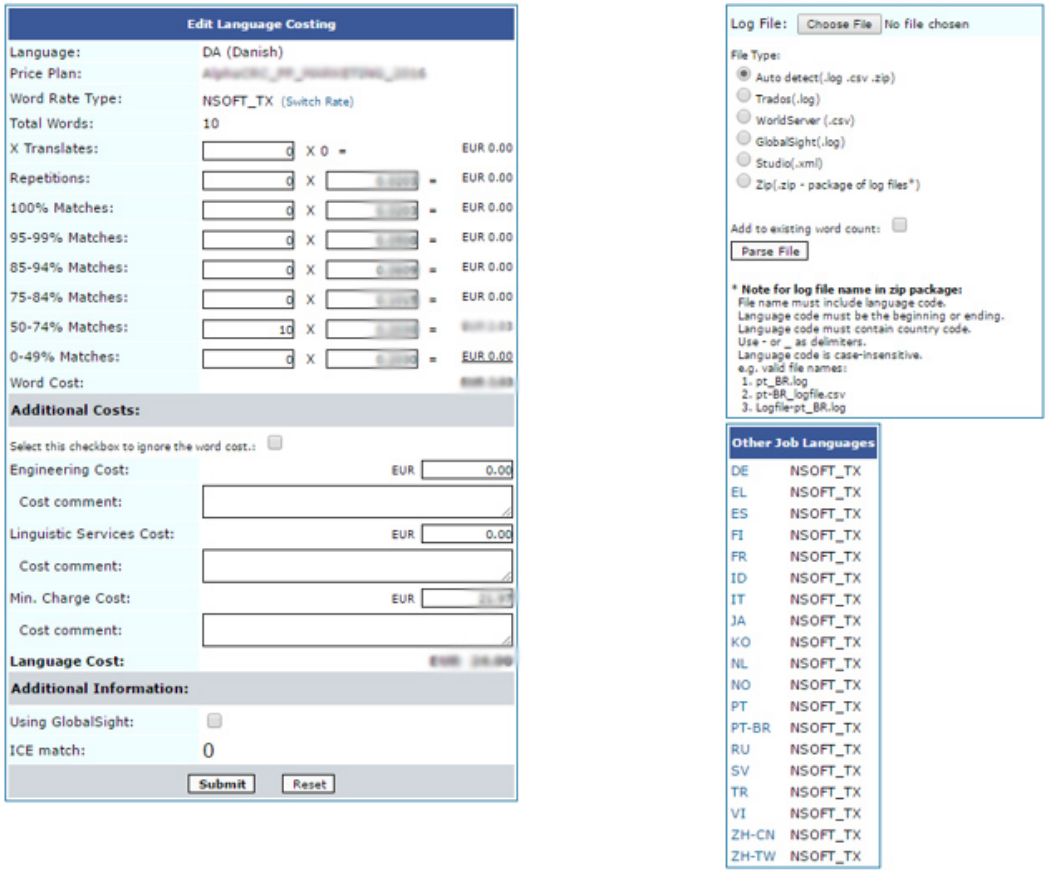


FIGURE 3.1: Existing Interface

- There is no REST API for connectivity with other internal tools
- The UI is dated and difficult to navigate around.

#### 3.1.1.1 Proposed UI Improvements

The outdated and unintuitive Fig 3.1 interface will be replaced with a modern, minimalistic interface using Bootstrap and AngularJs using familiar navigation and design patterns to visually aid the user in using the application as can be seen in the mockup in Fig 3.2.

#### 3.1.1.2 Problems with the existing data structure

The existing database in Fig 3.3 was built on top of an old bulletin board database and retains many extraneous tables relating to the management of posts and comments. The tables that are in use are not normalised in any way.

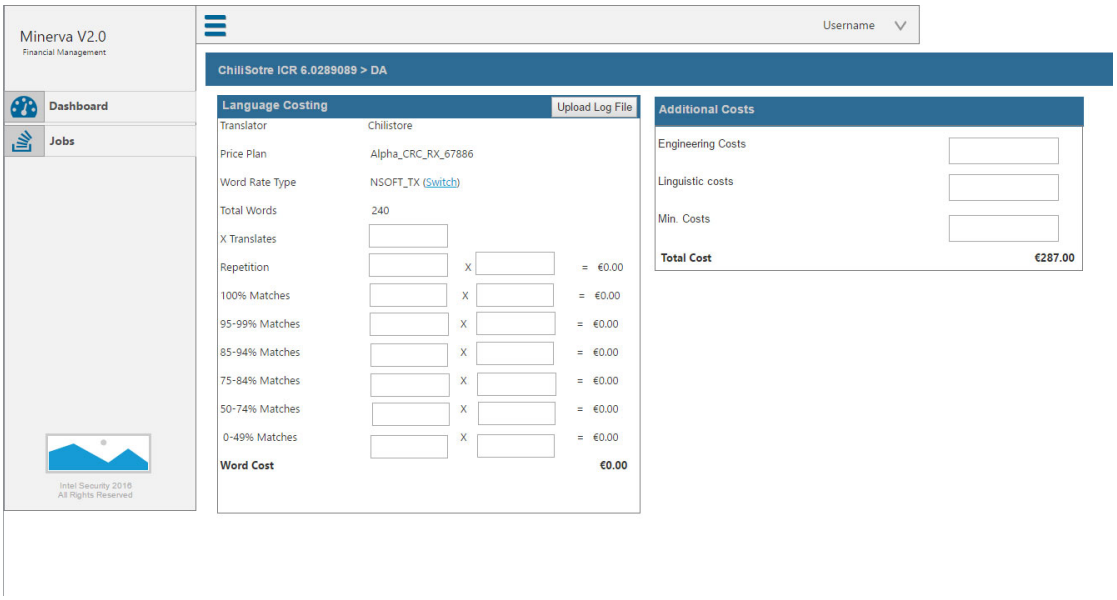


FIGURE 3.2: Proposed Interface Mockup

It appears from reverse engineering it that the system is reliant on a large amount of views to structure the data for processing.

Finally, sensitive data such as user password is stored as plain text which is a security risk.

#### 3.1.1.3 Proposed Structure

As shown in the first draft entity relationship model in Fig 3.4, I will normalise the data to its 3rd normal form by creating relationships and removing duplicate and redundant tables.

While some of the views may still be needed I will endeavour to reduce the need for them with good data structure from the ground up using logical connections.

It is likely that the server side code can be vastly improved and simplified resulting from these improvements.

### 3.1.2 Work-flow Issues

As touched on in the previous chapter, the conventional software translation work-flow is managed by translation management systems (TMS) such as GlobalSight. These systems are designed to leverage translation memories against uploaded files, assign


FIGURE 3.3: Existing Database Schema

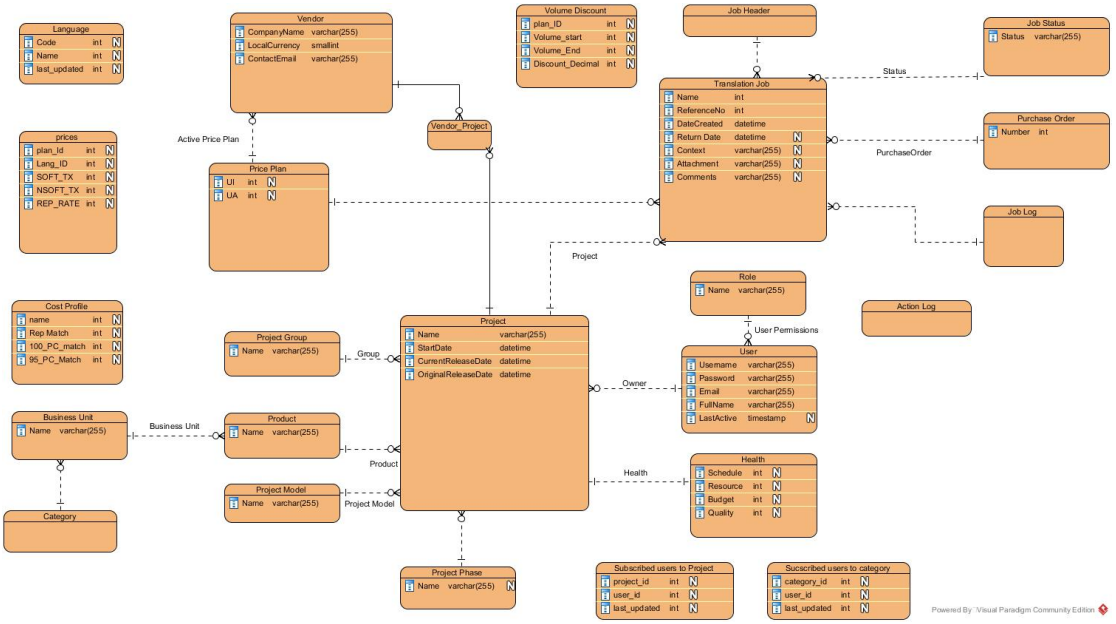


FIGURE 3.4: Proposed Domain Model

translators to jobs and generate financial quotes for clients. These systems, while effective, are designed to fit a company doing in house translation and fall short when it comes to managing outsourced service providers and providing a financial interface for project managers. This system will sit in-between the internal systems and the external GlobalSight to automate the process of assigning these jobs to vendors and calculating payment due upon job completion based on profiles created for the service providers.

#### 3.1.2.1 Current System

This use case assumes that the client profile for the vendor has already been created and all relevant rates for that vendor have been set up.

Fig 3.5 outlines the use case of a project manager creating a translation job in Translation Manager, one of Intel Securitys internal localisation tools.

The job is created in Translation manager, an internal application providing all relevant details of the job, the translation kits and any reference material such as screenshot URLS. During the job creation process, one or more a specific language service providers are assigned to work on it. Once completed, Translation Manager sends the job information both to the existing financial management system and GlobalSight. Once created, the project manager must manually send an email to notify the vendor of the assigned job who must them accept the job within GlobalSight. Once accepted, the project manager must change the job status from created to in progress.

The remainder of the work-flow happens within GlobalSight as translations are completed, re-viewed and eventually accepted. Once this process is completed, the vendor will upload the translation log files to the system and notify the project manager who will then change the status again to completed, review the financial information and arrange payment.

With this system, there are many manual stages involved in changing the project status and sending notification emails. These manual tasks could easily be forgotten leaving the project in an undesirable state and potentially costing the company money.

#### 3.1.2.2 Proposed System

Fig 3.6 shows the proposed work-flow for the new system which will automate the notification emails and the project status resulting in a streamlined system and removing much of the potential for error as a result.

## 3.2 Stakeholders

The stakeholders of the system are outlined in the requirements document provided to me at the beginning of this project and represent the main user roles that will be interacting with this system.

Each of the stakeholders were interviewed at their place of work and were asked to complete their regularly performed tasks with the existing systems. The steps they took to complete these tasks were monitored and any problems they had while doing them were noted.

After they had completed the tasks the user was asked if they had any issues with the current process and, if so, how they would like the new system to function to enable them to complete this task more efficiently.

### 3.2.1 Administrator

A localisation system administrator who is responsible for managing the user and project data within the localisation systems. They are familiar with engineering and administrative tasks.

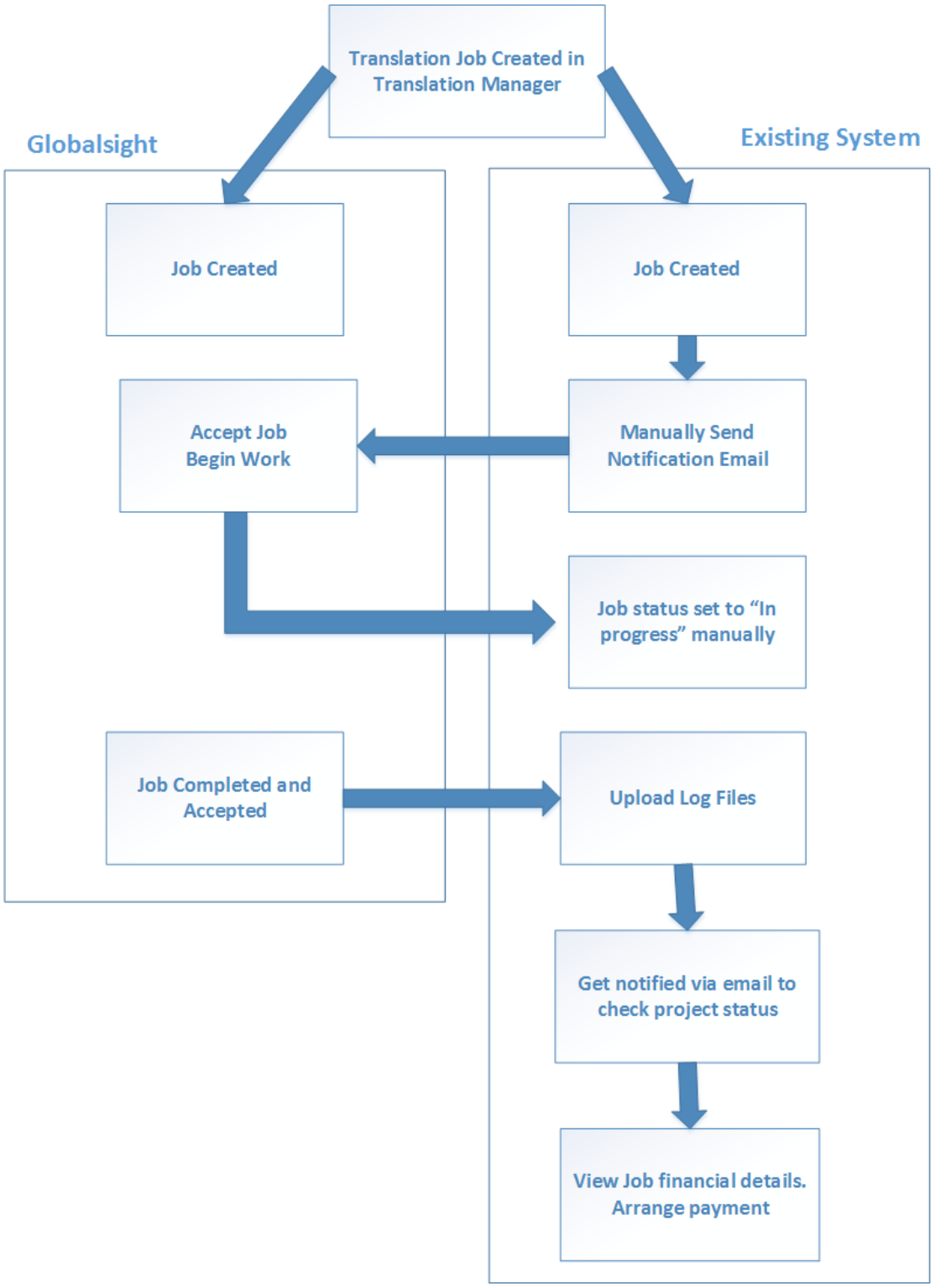


FIGURE 3.5: Current Work flow - Localisation Project manager

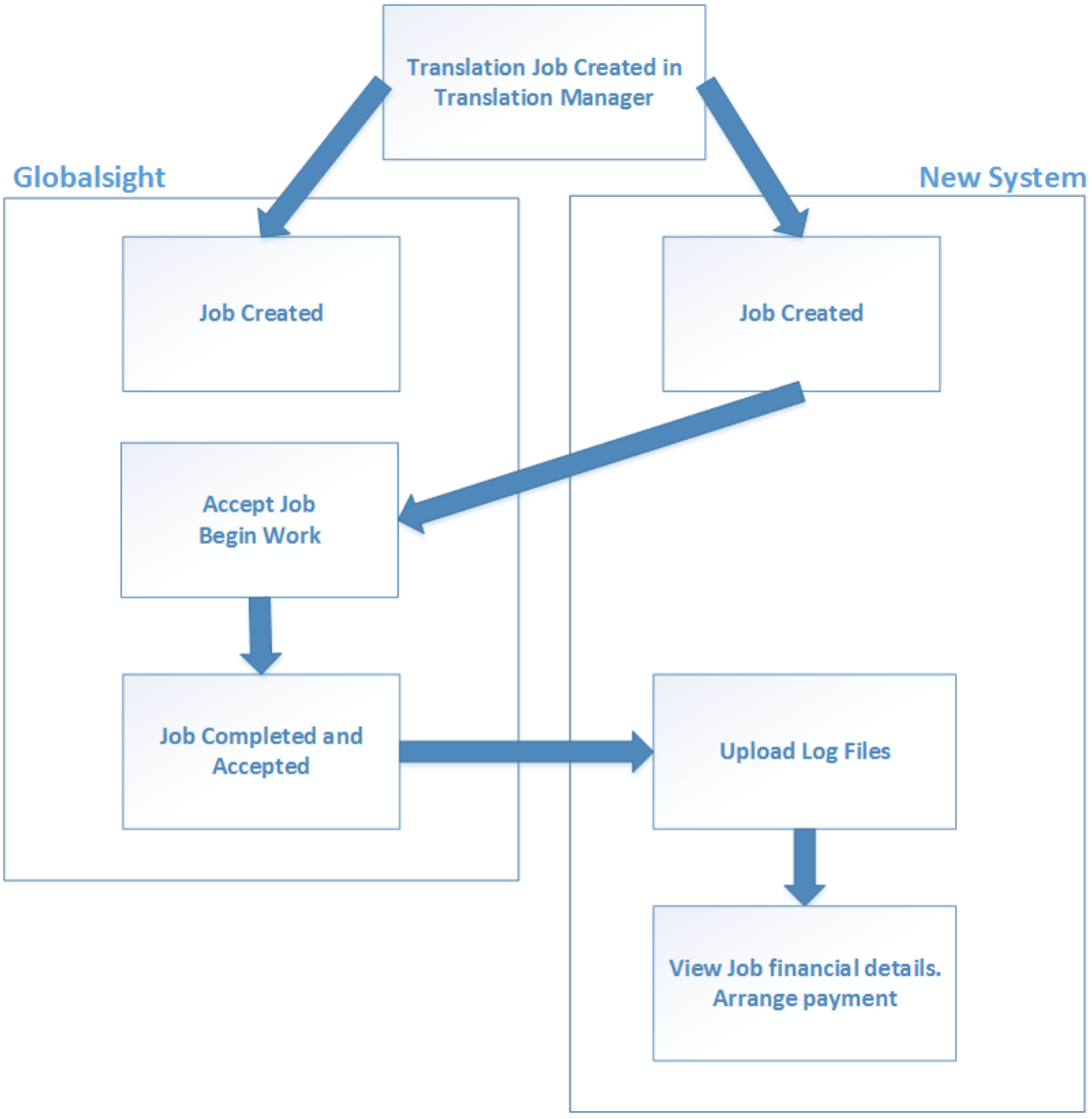


FIGURE 3.6: Proposed Work flow - Localisation Project manager

#### 3.2.1.1 Functional Requirements

The administrator will use the application to set up new user accounts and modify existing accounts. They need to be able to create accounts, look up accounts and easily modify account permissions and sometimes need to add and edit projects to the system.

#### 3.2.1.2 User Stories

- As an admin, I want to manage user accounts so that I can add new users to the system.
- As an admin, I want to be able to create a new project on the system so that translations jobs can be associated with it.

### 3.2.2 Program Manager

A software localisation project manager who is responsible for managing the project at a high level. They deal with the translation vendors, the software engineers and upper management so are a hub between them all.

#### 3.2.2.1 Functional Requirements

The Project Manager uses the system to view the status of projects and translation jobs, to assign financial data to these jobs and to calculate the payment due to the translation vendors once completed.

#### 3.2.2.2 User Stories

- As a Program Manager, I want to create and modify projects
- As a Program Manager, I want to see a summary of a project before submission
- As a Program Manager, I want to see a dashboard on Projects
- As a Program Manager, I want to filter data in the dashboard
- As a Program Manager, I want to search for specific projects
- As a Program Manager when I click on a project I want to see project specific data
- As a Program Manager, I want to see a translation request on the dashboard
- As a Program Manager, I want to see full details we store of the job (Outlined in Engineering Use Stories)

### 3.2.3 Engineer

A localisation software engineer is responsible for maintaining the various applications and frameworks that the localisation department uses.

#### 3.2.3.1 Functional Requirements

They use the system primarily via the API but frequently need to edit translation requests directly in the system.

#### 3.2.3.2 User Stories

- As an Engineer, I want to modify translation requests.
- As an Engineer, I want to view the details of a Submitted job.

### 3.2.4 Translation Vendor

A translation vendor is a company who provides translation services. There will be many of these vendors used for each software localisation project and they are located all over the world.

#### 3.2.4.1 Functional Requirements

They will use the system to create vendor accounts, projects and manage vendor price plans. They needs an intuitive system that allow them to set up accounts quickly and repeat the task with the same options afterwards to add multiple accounts. They also needs the ability to set up and modify the vendor price plans which detail how much each vendor chargers for their services.

#### 3.2.4.2 User Stories

- As a vendor, I want a dashboard that will allow me to view all jobs and languages assigned to me
- As a vendor, I want to change the status of multiple jobs at once
- As a vendor, I want to print or download a pdf roundup of all jobs set to receive for me
- As a vendor, I want to upload logs and have costs calculated automatically by the system.

### 3.2.5 Vendor Manager

A vendor manager is responsible for managing accounts of translation vendor companies in the system. They maintain the prices that these companies charge and calculate payment due from translation jobs

#### 3.2.5.1 Functional Requirements

The Project Manager will use the system to create vendor accounts, projects and manage the vendor price plans. They need an intuitive system that allow them to set up accounts quickly and repeat the task with the same options afterwards to add multiple accounts. They also needs the ability to set up and modify the vendor price plans which detail how much each vendor chargers for their services.

#### 3.2.5.2 User Stories

- As a vendor manager, I want to create and modify vendor companies.
- As a vendor manager, I want to upload, edit, make inactive or delete vendor price plans (also known as rate cards)
- As a vendor manager, I want to modify project categories.
- As a vendor manager, I want to modify pricing connections between the categories in the price plan and project categories.
- As a vendor manager, I want to manage costs for non-linguistic services (development, external QA, screenshot creation, etc.)
- As a vendor manager, I want to take account of Volume discount rates automatically once the threshold has been reached.

## 3.3 Non-Functional Requirements

Non-functional requirements define constraints on the way the software-to-be should satisfy its functional requirements or on the way it should be developed.

### 3.3.1 Consistent Design

To be consistent with other software in use by Intel Security, a style guide will be used when implementing the web application to ensure it is consistent. The company provided this guide and it can be seen in Table 3.1.

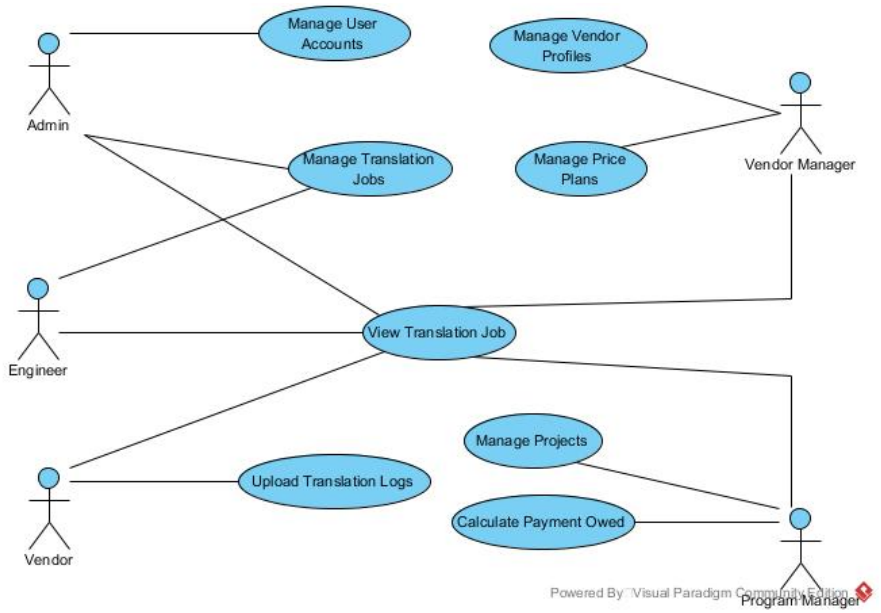


FIGURE 3.7: Use Case Diagram

TABLE 3.1: Style Guide

| **Content Type** | **Style Setting** |
|---|---|
| Font Family | Intel Clear, Sans-Serif |
| Font Colour | #53565A |
| Body Font Size | 12px |
| Heading 1 Size | 21px |
| Heading 2 Size | 18px |
| Heading 3 Size | 16px |
| Heading 4 Size | 14px |
| Accent Colour | #0071C5 |
| Accent Accent Colour | #7fd3f7 |

### 3.3.2 Nielsen's principles for interaction design

Nielsen defines high level principles for interaction design in "Heuristic Evaluation of User Interface" which provide a set of heuristics with maximum explanatory power[15].

- Visibility of system status
- Match between system and the real world
- User control and freedom

- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

Following these principles will ensure that the user experience is a simple and intuitive as possible resulting in a more robust final product.

### 3.3.3 Speed and Latency

The interface and the underlying logical operations must respond with a maximum response time of 1 second. If the system fails to respond within this time frame for any reason beyond its control the user should be notified of the issue and presented with an option to try again.

### 3.3.4 Security

The application will run within secure Intel systems with enterprise level security preventing unauthorised access.

### 3.3.5 Maintainability

The code-base will be secured on private Git servers and commits will be monitored by a build system ensuring that the updated code passes all regression tests and, if not, the source of the problem can be easily identified.

### 3.3.6 Easy of Use

The application must allow the user to complete their core tasks with a minimum of 4 clicks. This will be accomplished by using familiar design patterns such as the dashboard pattern ensuring that core information and tasks are presented to the user upon login without the need to navigate to other sections.

### 3.3.7 Design Patterns

Design patterns are commonly found patterns in interface design which reply on previous user experience with other applications and intuitive positioning to provide the user with a better experience.

#### 3.3.7.1 Dashboard

Dashboards are a common design pattern used throughout web applications and enable the user to consume data from many sources on a single page. They are often customizable by the user or set based on a user role to ensure that only relevant information is displayed[16].

This design pattern will be implemented for all users of the system and will be their landing page after they log into the system. The goal will be to allow them to complete all of their regular tasks from the dashboard without the need to navigate deeper into the application and fulfill the User control and freedom heuristic.

#### 3.3.7.2 Breadcrumbs

Breadcrumbs are a sequence of links which indicate to the user their hierarchical location within the application and allow them to navigate to higher level pages. They are an effective visual aid and serve to orientate the user especially when landing on a page deeper in the hierarchy[17]. They are traditionally separated by a delimiting character indicating the connection between the links.

This design pattern will be implemented on all non-top level pages and will be especially useful in sections with deeper navigation such as job detail views and language views and will fulfill the visibility of system status heuristic.

#### 3.3.7.3 Accordion Menus

Accordion menus are collapsible links which, when clicked, display subsections of the selected section enabling the user to minimise the information on screen when it is not needed and adhere to the aesthetic and minimal design heuristic[18].

#### 3.3.7.4 Progressive Disclosure

Progressive disclosure involves hiding information until it is immediately important to the user thus giving them less choice and resulting in less confusion during complex tasks[19].

This will be implemented in the project creation systems by using progress bars and sectioning off the content needed into related sections such as basic information, vendor details and language details.

# Chapter 4

# Implementation Approach

## 4.1 Guidelines Set by Intel

The application will be developed to the specifications of Intel Security using technologies that their development team is familiar with and that their other applications are developed with. This will ensure that the application can be maintained after it is completed by myself and that it is as compatible as possible with existing systems.

The technologies used are ones that I am familiar with from previous experience working with Intel and from my studies so, while the choices were somewhat mandatory, I am comfortable developing with these technologies.

## 4.2 Architecture

As per the specification, the application will comprise of 3 main components. The client side development will be done using JavaScript and the AngularJS framework, the server side development which will manage the data and expose an API to the client will be developed using C# and the underlying database will be created in SQL Server as seen in Fig 4.1.

### 4.2.1 The Client

The client is what the user of the application sees and interacts with, it is rendered and displayed in a web browser and must provide a robust and intuitive way of interacting with the application.

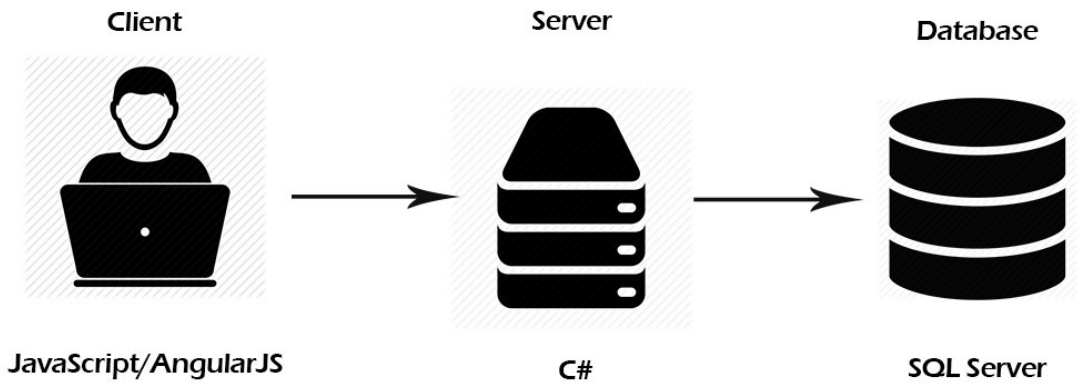


FIGURE 4.1: Application Architecture

#### 4.2.1.1 JavaScript

JavaScript is a simple and effective language which is used to extend the functionality of basic websites alongside HTML and CSS and, in many cases used to entirely develop web applications. It is executed on the client site in a browser which reduces server load and enables relatively rapid feedback within the interface[20].

Using pure JavaScript to develop a complex web application however can result in messy and disorganised code so to fully realise an application in JavaScript a framework is frequently required.

#### 4.2.1.2 AngularJS

AngularJS is a structural JavaScript framework developed by Google for developing web applications. It allows the use of HTML templating to extend the functionality of basic HTML pages and simultaneously integrate them with the underlying JavaScript code making for much smoother integration with server side technologies[21].

This is achieved by using a Model View Controller (MVC) pattern to separate functionality of the code into sections depending on their function. The model code deals with Create Read, Update and Delete (CRUD) operations by interfacing with the database usually via an API. The view is the HTML markup that will render the application in the browser for the user to see and the controller which binds the above two together handling all of the logic in between, see Fig 4.2.

AngularJS extends upon this pattern by further modularising code into services and directives. Services can be used by a controller to interface with a model code and

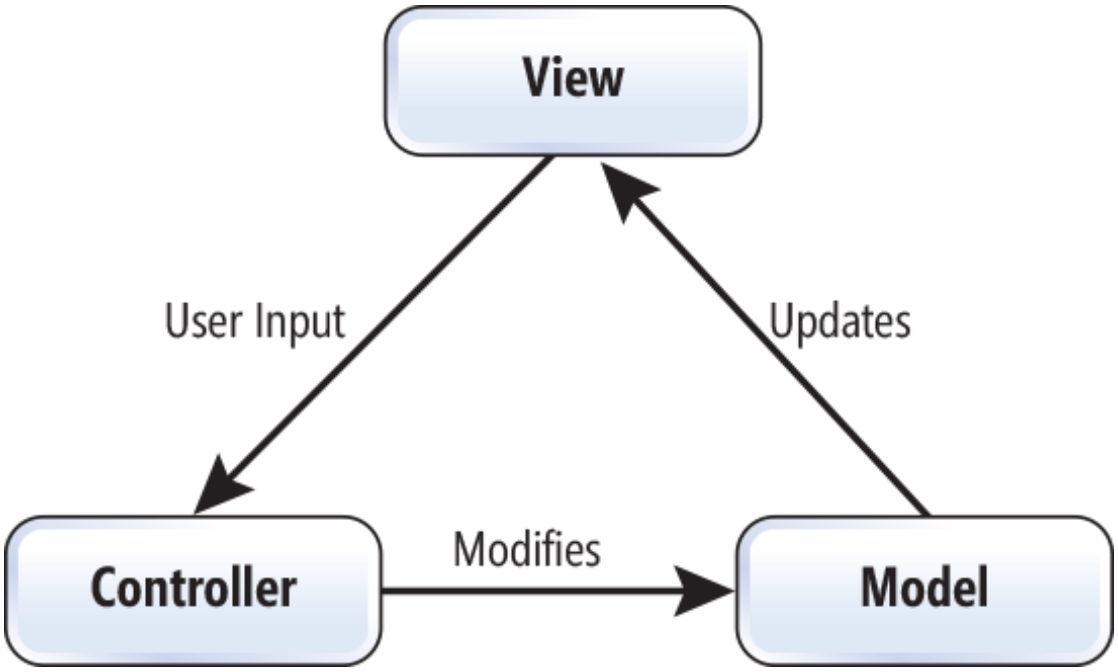


FIGURE 4.2: MVC Pattern

directives are fragments of HTML. This allows for greater maintainability and multiple controllers/views can make use of these without the need for duplication.

There are many libraries which can be easily added to an application to add functionality such as form validation or HTTP requests without the need to develop the functionality for those on your own.

Due to the popularity of AngularJS, there is a large community of developers and documentation online to provide any support that may be needed for this project along with the large number of libraries.

#### 4.2.1.3 Bootstrap

To reduce the development needed to create a responsible and highly compatible interface, the Bootstrap framework will be used. This framework arranges content in a grid system and allows for the page to respond to varying screen sizes by changing the layout of the elements as needed as seen in figure 4.3. Bootstrap also provides a number of built in icons, fonts and UI elements such as buttons and image carousels[22].

FIGURE 4.3: Bootstrap grid

CSS

```
1  #menu {
2    margin: 0;
3    list-style: none;
4  }
5
6  #menu li {
7    float: left;
8  }
9
10 #menu li a {
11   display: block;
12   float: left;
13   padding: 4px 8px;
14   text-decoration: none;
15   background: #2277aa;
16   color: white;
17 }
```

SASS

```
1  $menu_bg: #2277aa
2
3  #menu
4    margin: 0
5    list-style: none
6    li
7      float: left
8      a
9        display: block
10       float: left
11       padding: 4px 8px
12       text-decoration: none
13       color: white
14       background: $menu_bg
```

FIGURE 4.4: Sass vs CSS

#### 4.2.1.4 Sass/Compass

Syntactically Awesome Stylesheets (Sass) is a CSS extension language which allows the use of variables for consistent styling and mixins to make CSS more manageable for large projects[23]. A mixin is a class which contains functions and variables that other classes can use such as a library in conventional programming. Here is allows for modular CSS code which can take advantage of the large number of prepared mixins to reduce development time. The use of variables and the difference in between Sass and CSS can be seen in Fig4.4 taken from Smashing Buzz [24].

For this project Sass will enable me to use predefined CSS properties from the provided style guide to ensure a consistent style across the application. The modular mature of

Sass will also allow me to reuse any existing classes that may be in use in Intel's other systems to reduce development time.

As Sass is not standard CSS, it is not understood by web browser and must first be converted to standard CSS before running the application. This is done with a Ruby gem called Compass[25] which will run as part of the Gulp script which builds the application for testing and for production.

#### 4.2.1.5 Gulp

Gulp is a JavaScript task runner which automates build tasks for complex web applications[26]. Scripts will be created which will ensure that all of the applications dependencies and source files are loaded correctly and converted into production ready, often minified versions.

Gulp is a module written in Node.js and provided with the node package manager(Npm) system and will ensure that the application build tasks run correctly and notify me of any issues prior to testing.

#### 4.2.1.6 Node/Npm

Running on top of the server side JavaScript environment Node.js[27], node package manager allow me to easily load modules such as the above onto my development environment and as dependencies for the application.

This will enable me to leverage powerful external libraries to further reduce development time.

### 4.2.2 The Server

The server side development will be done using Microsoft's .NET framework which incorporates C# and SQL server to create cross platform web applications in a Windows server environment[28].

#### 4.2.2.1 C#

C# is an object oriented programming language which is intended to be simple and modern. It's solid integration with Visual Studio[29] allows for rapid development with

a large selection of packages available and it's ability to scaffold out commonly used coding components.

#### 4.2.2.2 SQL Server

SQL Server is Microsoft's database offering that is integrated with the .NET framework allowing ease of use within a C# application[30]. Unlike its main competitor MySQL, is it not open source but rather a proprietary Microsoft solution. This results in it being very secure and stable but at financial cost.

#### 4.2.2.3 Linq To SQL

Linq to SQL is a component of the .NET framework which allows for data modelling and object relational mapping (ORM). It translates database result sets into objects which can be used within object oriented code and form objects into SQL statements for modifying the database. This results in easier to read code which follows the conventions of object oriented programming more so than wrapping SQL statements within your programming code.

### 4.2.3 Testing

As the application is being developed, end to end tests will be written to ensure that modules functionality are not negatively affected by future changes to the code base. The testing will consist of unit tests to ensure that the logic of the code, where applicable, works as expected and interface tests which will automate a web browser and simulate user interactions to ensure that the expected behaviour takes place.

#### 4.2.3.1 Jasmine

Jasmine will be used for this test code as it is written in JavaScript as is the client side code it will be testing[31]. Tests written in Jasmine as easily readable and the output is descriptive as it provides more verbose errors telling the developer where the problem is in English rather than a complex stack trace.

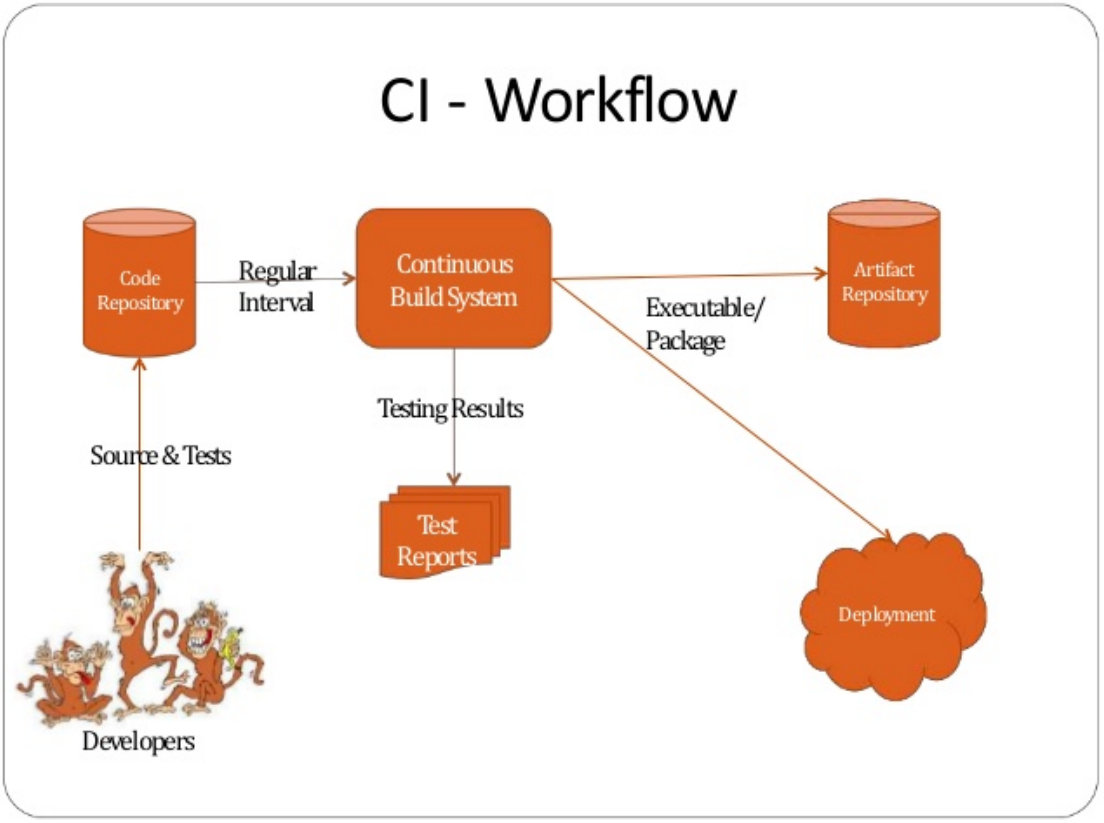


FIGURE 4.5: Continuous Integration

#### 4.2.3.2 Selenium

Selenium is the software which will be used to automate the web browser. The above test code will connect to a locally running Selenium instance and issue commands to the browser via selenium which will simulate the actions of a user.

### 4.2.4 Continuous Integration

Continuous integration is a development practice whereby code is stored on a centralised repository, in this case GIT, and all commits are handled and verified by an automated build system, in this case Jenkins. This practice ensures that all new and modified code passes predefined tests and does not negatively impact the rest of the system.

This process is visualised in Fig 4.5 [32].

#### 4.2.4.1 Git

Git is a version control system which tracks changes to files enabling multiple people to work on the same files simultaneously and providing functionality for merging those changes together once complete. Git also maintains a log of all changes made and allows reverting to previous versions should anything go wrong.

#### 4.2.4.2 Jenkins

Jenkins is an automation server which provides functionality for building and deploying software projects. Along with Git it will form the core of the continuous integrations system to ensure that all builds are testes and stable before release.

## 4.3 Risk Assessment

Performing a risk assessment can identify potential problems that may arise within an application. During this assessment a list of possible risks is identified and the potential impact on the project is calculated along with a solution where possible.

### 4.3.1 User Risks

This section details the potential risks to the user experience.

#### 4.3.1.1 Minor - Font unsupported by client browser

The users web browser may not have the Intel Clear font installed resulting in a less than optimal aesthetic should it default to another font. This can be resolved by specifying a basic font to default if Intel Clear is not found on the system.

#### 4.3.1.2 Major - Client screen resolution too small

The users monitor or operating system resolution settings may be too small to correctly display all of the content on the page causing content to overlap and become inaccessible to the user.

This can be prevented by leveraging the Bootstrap front end framework to make the page responsive and account for even mobile phone based browsers.

#### 4.3.1.3 Critical - CSS Property unsupported

Some older and less popular browsers may not support all of the advanced styling that is required to properly display the page resulting in problems. This can be accounted for to some extent by using JavaScript libraries such as modernizer and HTMLshiv which use scripting to interpret CSS properties that are unknown to the browser.

#### 4.3.1.4 Fatal - JavaScript Unsupported by browser

Is it possible that the user's browser may not support JavaScript at all which would render the application inert. There is no way to handle this but the user can be properly notified of the issue by use of ¡noscript¿ tags which are displayed to the browser is JavaScript cannot be run.

### 4.3.2 Server Risks

This section details the potential risks to the server side code and environment.

#### 4.3.2.1 Minor - Concurrency causes minor increase in latency

A large number of users accessing the application at the same time may cause the system to slow down slightly. This is to be expected and thresholds can be set in the form of timeouts in the code which will allow for a certain delay before returning an error to the client.

#### 4.3.2.2 Major - Concurrency causes unacceptable increase in latency

In the case where the number of users causes a large increase in latency beyond an allowable threshold there must be limits put on server use. One solution to this would be to limit the number of active users and to display a message to other users that they must wait before logging in.

#### 4.3.2.3 Critical - Server IP changes

The server IP address may change if the DHCP lease expires or if the server machine is moved to another location. This would cause issues in domain name resolution which map the URL to the IP. The solution to this is to map the physical (MAC) address of the machine to a specific IP so it retains the same IP wherever it is on the network.

#### 4.3.2.4 Fatal - Server Offline

Server outages are inevitable and mostly beyond control however they must be accounted for in the client side code. If the server takes too long or cannot be connected to at all, an appropriate message must be displayed to the user and the code must handle it to prevent a crash.

### 4.3.3 Database Risks

This section details the potential risks to the database.

#### 4.3.3.1 Minor - Excessive allocation given to fields

If a large allocation is given to small data fields, for example a "varchar(250)" being assigned to a field which will only store a zero or a one, over time this wasted disk space can build up and potentially cost the company money. This can be prevented by correctly planning the database and the data types it will store.

#### 4.3.3.2 Major - Incorrect data types

Incorrect data typing, similarly to the above can result in wasted disk space but also may present unforeseen issues later in the development cycle. This can again be prevented by correct planning.

#### 4.3.3.3 Critical - Un-Sanitized strings

Before saving a string into the database it must first be sanitized by the code. This process removes any special characters from the strong or escapes them such that the server side or database code does not misinterpret them as commands and present a security and data integrity risk. JavaScript and C# both have robust sanitation libraries which will be implemented in this project.

#### 4.3.3.4 Fatal - Password Truncated

If the max possible characters in the client side password input box is larger than the max possible characters that can be stored in that field in the database it is possible that the database will try to store the string anyway by truncating off the excess characters.

Often this will not affect the user but, depending on how the credentials are handled in the code it may result in a wrong password error as they do not match. This can be prevented by ensuring that the form validation matches the data types in the database.

### 4.3.4 Security Risks

This section details the potential security risks.

#### 4.3.4.1 Minor - SSL Certificate out of date

SSL certificates allow for HTTPS connections but can often go out of date resulting in errors in the client. This is usually just a pop up which asks the user to accept the out of date certificate but could cause other complications. This can be prevented by ensuring that all certificates are kept up to date.

#### 4.3.4.2 Major - Un-Sanitized strings

As with the above database risk, un-sanitized strings can pose a security threat which can be prevented with sanitation libraries.

#### 4.3.4.3 Critical - Password saved as plain text

Simply saving a users password in the database as a text field presents a security issue should anyone gain access to the database and is bad practice. To prevent this the password should be hashed using MD5 or any other encryption method to ensure correct security.

#### 4.3.4.4 Fatal - Unencrypted credentials

When authenticating with the system it is inevitable that user credentials will need to be transmitted to the server. If they are transmitted simply as they are, unencrypted, anyone that intercepts the data along the way would be able to view them. To prevent this, HTTPS or another method of encryption must be used.

| | Minor | Major | Critical | Fatal |
|---|---|---|---|---|
| User Experience | Font unsupported by client browser | Client screen resolution too small to render page correctly | CSS property unsupported by browser | JavaScript Unsupported by browser |
| Server | High concurrency causes minor increase in latency | High concurrency increases latency to unacceptable levels | Server IP changes. DNS issues | Server is offline due to a crash or IT related issue |
| Database | Large allocation given for small data fields | Incorrect data type for fields | Un-Sanitized strings saved into database | Password truncated due to max length when saved |
| Security | SSL cert out of date | Data not sanitized before saving into database | Password saved as plain text | Unencrypted credentials transferred across network |

FIGURE 4.6: Risk Assessment Matrix

## 4.4 Methodology

The methodology with which one develops software can have a large effect on the end product. In the past, the waterfall method has been the preferred method which involved a lengthy requirements specification at the start of development and the end product being delivered based on that. This often resulted in software being produced that simply did not meet the stakeholders requirements as the requirements may have been misunderstood or may have changed thought a long development cycle. To combat this, and ensure that valuable software is produced, the Agile methodology will be used for this project.

### 4.4.1 Agile

This is a set of principles for software development which values short development cycles and working software and collaboration with stakeholders at all stages of development.

The following are the core principles as defined in the Agile manifesto[33].

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.

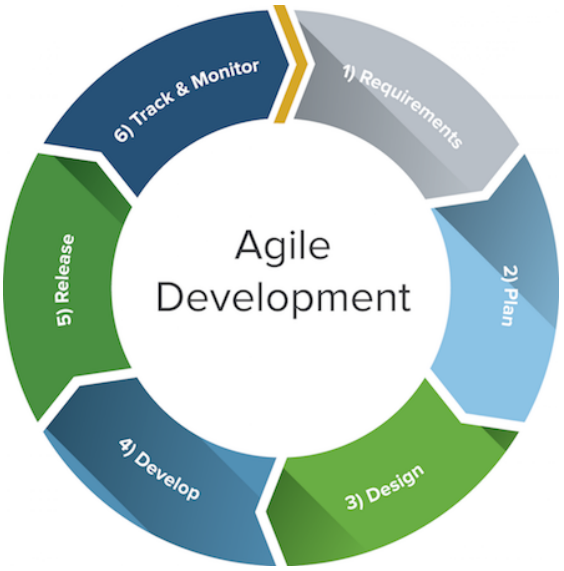


FIGURE 4.7: Agile Lifecycle

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity–the art of maximizing the amount of work not done–is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

In accordance with these principles short development iterations will be organised. During these iterations of "Sprints" one or more of the user stories from the requirements spec will be assigned to me and the work required to complete this story will be planned. At the end of the sprint all of the functionality required to fulfill that user story will be completed and ready to release and fully tested to ensure compatibility and robustness. At all stages throughout, collaboration with all stakeholders will be crucial to ensuing that they get the end result that they want as seen in Fig 4.5.

### 4.4.2 Modular Code

Modular programming involves separating out code by its function so that each of the modules encapsulates all of the code to complete its functionality. This allows modules to be reused and independently tested and makes for a higher degree of maintainability and reduces development time.

### 4.4.3 Secure Code

The application and it's code base will be housed on secure Intel Security servers which will ensure that it is protected from unauthorised access by a dedicated IT team. however, the code itself must be developed with security in mind and follow secure coding practices[34].

1. Data validation must be done on trusted systems to prevent data tampering
2. Input validation and sanitation to prevent SQL injects and buffer overflow attacks
3. Data in transit must be encrypted to prevent interception

## 4.5 Implementation Plan Schedule

The implementation of this application will be done in two week sprints from January to May which will divide the months in two evenly as can be seen in Fig 4.6.

The actual content of these sprints will be decided at a later date following a sprint planning session with the team at Intel. As such, I have provided a general overview of the components to be completed with a possible time-line for their completion.

The design of the database will be completed in January after finalising the schema with my manager at Intel and it will be implemented in SQL server based on this design by the end of January in parallel with the start of the server side component implementation such that they can be tested with one another.

The server side component will be a large project which will be separated into many sprints, each focusing on a different segment of functionality which will be decided up on in sprint planning sessions. This component is due to be completed at the end of March but will likely be modified after that as issues arise with the client side.

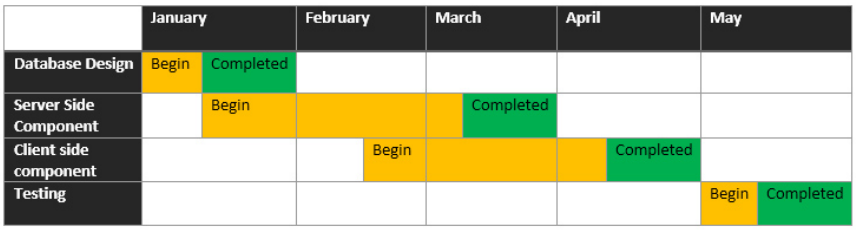


FIGURE 4.8: Implementation Plan

Development on the client side component will begin mid February and be completed at the end of April. Like the server side component this will be a large project and will be separated into many sprints.

Once all of these are completed development of the testing suite will begin and as much of the code base as possible will be covered by the time the project is due to be presented in May.

## 4.6 Prototype

A prototype of the application was created in Axure and tested in order to gain insight into potential problems before implementation.

### 4.6.1 Testing

The prototype was used with volunteers to test the functionality of the 6 tasks outlines below. They used scenarios based on stakeholders for the system and performed their tasks under this guise.

Their scenarios were based on the following use case and were as follows:

- Joe is a system admin and needs to add an account for a newly hired engineer. The engineers name is David, his email is david@intel.com and, using an external secure password generator his password will be set to TestPassword01.
- John is a project manager and wants to add a new project. The project is called ChilistoreIDC3829803" using the marketing model in the translation category for the FRP product using the default BU of EPO. It is to have an RTW date of 10/12/2016. ChiliStore must be assigned to the project along with DA and FR languages.

- Michael is a vendor manager and needs to add a new vendor account for LangMax using the email info@langmax.com using a securely generated password of Password01
- John is a project manager and wants to view the details of the project EPOMODULE
- Mary is a translator for a translation vendor. She wants to upload the log file of a translation job called App Store MMS iOS for the DA language.
- Paul is a software localisation engineer who wants to add GR as a language to the job "App Store iOS MMS"

Based on this testing feedback was given which will be helpful in the implementation phase of the project and exposed potential issues with the interface. Appendix B contains some relevant screenshots from the prototype showing the layout and interactions that were done during the testing.

# Chapter 5

# Conclusion and Future Work

## 5.1 Conclusion

With the specification of the project being laid out prior to starting the research phase it was a struggle to make it fit into the conventional format but the process has provided insight into the problems being solved and, with the requirements set by Intel the path to implementation is clear.

To correctly implement all of the features to the standard required there will be much work ahead and much learning to do but working on an enterprise level application will prepare me well for a career in web development.

## 5.2 Future Work

As the application will be designed in a modular and maintainable manner using technologies that Intel staff are familiar with , it will be simple to extend the functionality should they require it.

Were this project to be developed further by myself I believe that a focus on further improving the work-flow by adding increased automation and integration with existing systems.

# Bibliography


[1] Software localization - edutech wiki. [Online]. Available: http://edutechwiki.unige.ch/en/software_localization

[2] (2016) New to globalsight? [Online]. Available: http://www.globalsight.com/

[3] "What is salesforce? cloud crm solutions - salesforce europe," http://www.salesforce.com/eu/crm/what-is-salesforce/, (Accessed on 11/20/2016).

[4] "The agile movement," http://agilemethodology.org/, (Accessed on 11/20/2016).

[5] "ch02.pdf," http://www.infoq.com/resource/articles/scaling-software-agility/en/resources/ch02.pdf, (Accessed on 11/20/2016).

[6] "Enterprise agile lifecycle management (alm) solutions," https://www.versionone.com/, (Accessed on 11/20/2016).

[7] "Erp system — enterprise resource planning — sap," http://go.sap.com/uk/product/enterprise-management/erp.html, (Accessed on 11/20/2016).

[8] "Sap company history," http://go.sap.com/corporate/en/company/history.html, (Accessed on 11/20/2016).

[9] D. Miller, "Off-the-shelf vs. bespoke software development - alberon ltd," https://www.alberon.co.uk/blog/off-the-shelf-vs-bespoke-software-development, 02 2015, (Accessed on 11/20/2016).

[10] "Teach-ict ocr a2 ict g063 syllabus: custom bespoke and off the shelf software," http://www.teach-ict.com/as_a2_ict_new/ocr/A2_G063/335_implementing_systems/implement_custom/miniweb/pg4.htm, (Accessed on 11/20/2016).

[11] N. Reid. Cancel reply. [Online]. Available: https://www.ibm.com/blogs/cloud-computing/2014/04/5-considerations-choosing-saas-premises/

[12] Z. Iqbal, "Outsourcing: A review of trends, winners losers and future directions," http://ijbssnet.com/journals/Vol_4_No_8_Special_Issue_July_2013/9.pdf, 07 2013, (Accessed on 11/20/2016).

[13] "Localization and translation services from simultrans, a leading language service provider," http://www.simultrans.com/#why-simultrans, (Accessed on 11/20/2016).

[14] A. Smith, "viewcontent.cgi," http://digitalscholarship.unlv.edu/cgi/viewcontent.cgi?article=2479&context=thesesdissertations, 08 2012, (Accessed on 11/20/2016).

[15] Browse by topic and author. [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/

[16] A. Halsall. (2015) How to design effective information dashboards for your business. [Online]. Available: http://ux.walkme.com/design-effective-information-dashboards-business/

[17] J. Gube. Breadcrumbs in web design: Examples and best practices. [Online]. Available: https://www.smashingmagazine.com/2009/03/breadcrumbs-in-web-design-examples-and-best-practices/

[18] J. Rocheleau. (2016) Best practices for accordion menu in web design. [Online]. Available: https://webdesignledger.com/best-practices-accordions-in-web-design/

[19] A. Toxboe and contributors. (2016) Patterns education. [Online]. Available: http://ui-patterns.com/patterns/ProgressiveDisclosure

[20] (2016) Javascript — mdn. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript

[21] (2016) Angularjs superheroic javascript mvw framework. [Online]. Available: https://angularjs.org/

[22] J. T. Mark Otto and B. contributors. (2016) Designed for everyone, everywhere. [Online]. Available: http://getbootstrap.com/

[23] (2016) File: $Sass_referencesassdocumentation.[Online].Available$ :

(2016) The-sass-syntax.jpg (jpeg image, 600 382 pixels). [Online]. Available: `http://www.smashingbuzz.com/wp-content/uploads/2014/06/The-Sass-syntax.jpg`

(2016) Compass home — compass documentation. [Online]. Available: `http://compass-style.org/`

(2016) Easy to use. [Online]. Available: `http://gulpjs.com/`

(2016) Build amazing things. [Online]. Available: `https://www.npmjs.com/`

V. P. o. A. . E. S. David Fuller. (2016) .net - powerful open source cross platform development. [Online]. Available: https://www.microsoft.com/net

giorgiap. (2016) Any developer, any app, any platform. [Online]. Available: https://www.visualstudio.com/

(2016) Built to help you do more. [Online]. Available: https://www.microsoft.com/en-us/sql-server/sql-server-2016

(2016) Jasmine documentation. [Online]. Available: https://jasmine.github.io/

(2016) jenkins-build-system-6-638.jpg (jpeg image, 638 479 pixels). [Online]. Available: http://image.slidesharecdn.com/jenkins-160215171658/95/jenkins-build-system-6-638.jpg?cb=1455556649

(2016) Principles behind the agile manifesto. [Online]. Available: http://agilemanifesto.org/principles.html

(2016) Security by design principles - owasp. [Online]. Available: https://www.owasp.org/index.php/Security_by_Design_Principles

# Appendix A

# Wireframe Models

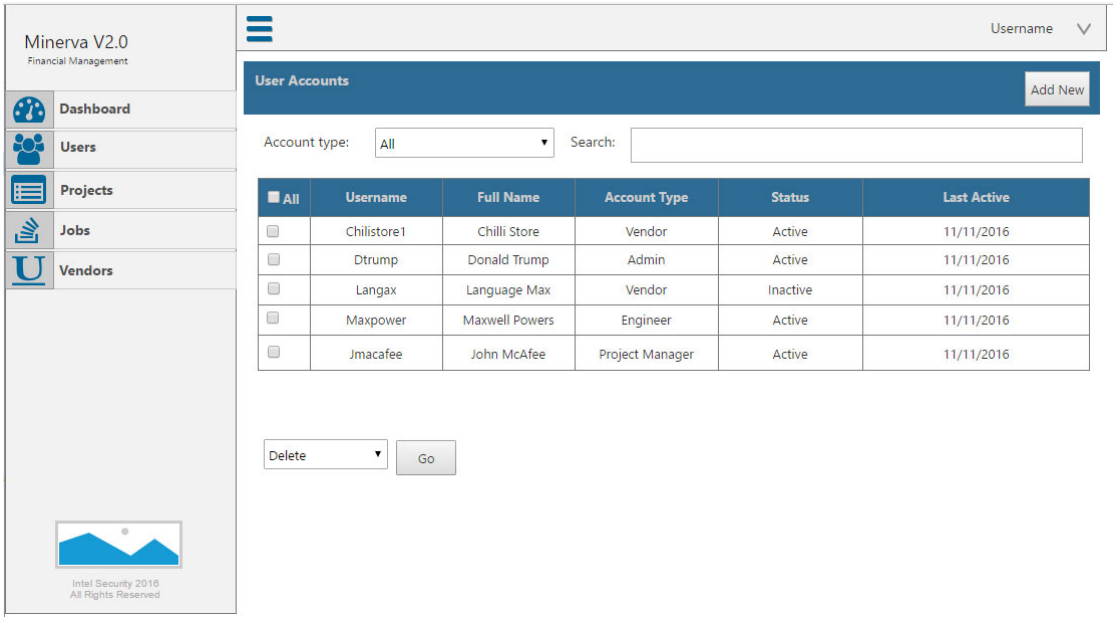


FIGURE A.1: Admin Dashboard

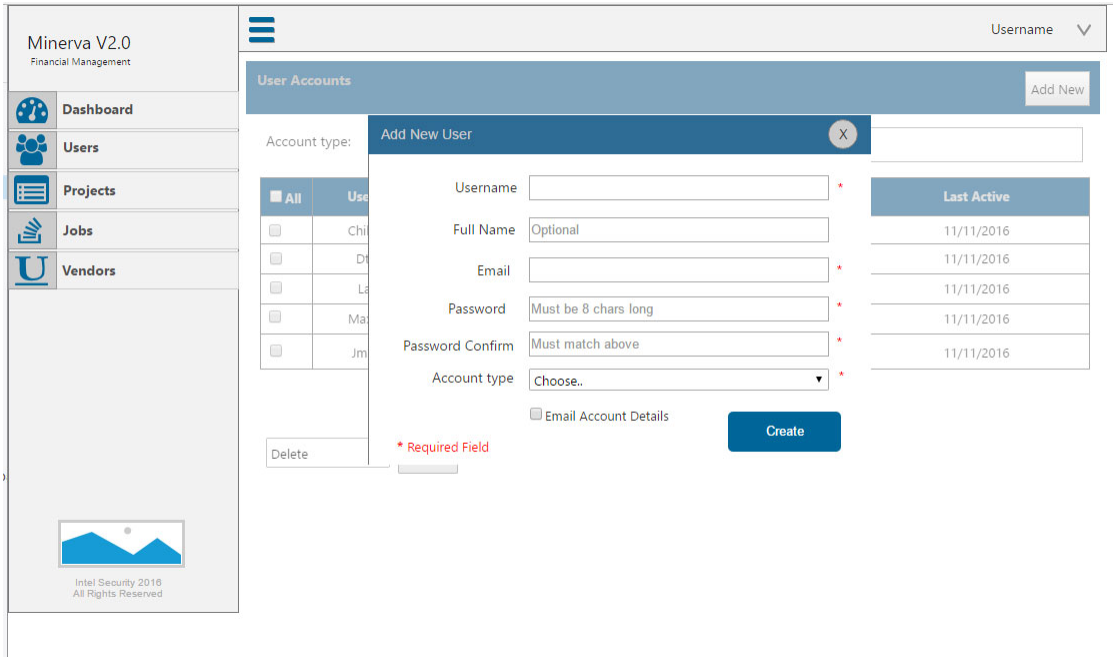


FIGURE A.2: Admin Add User Modal

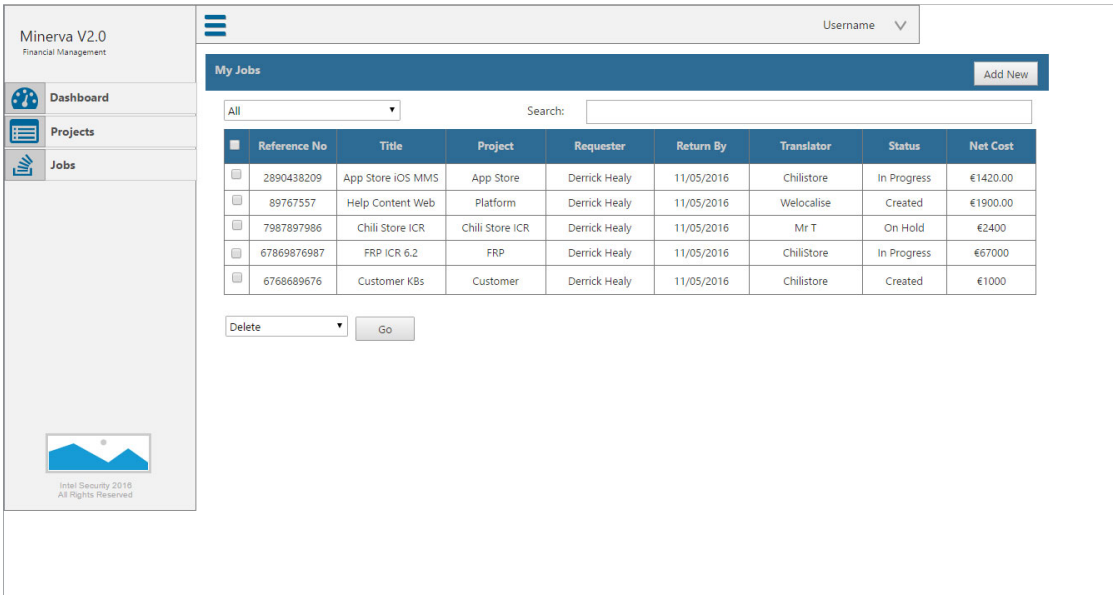


FIGURE A.3: Engineer 1

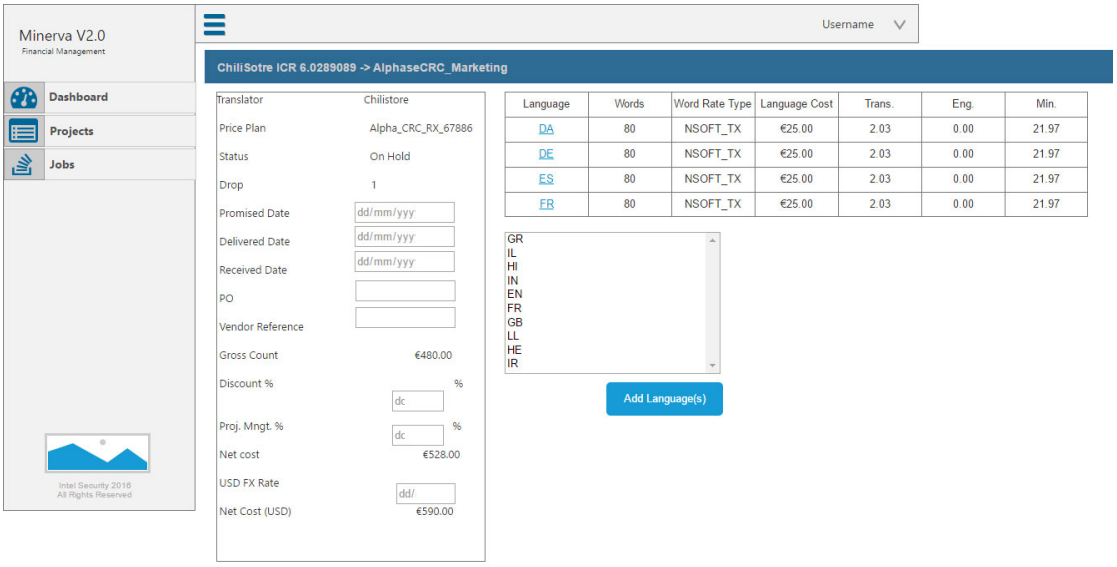


FIGURE A.4: Engineer Dashboard

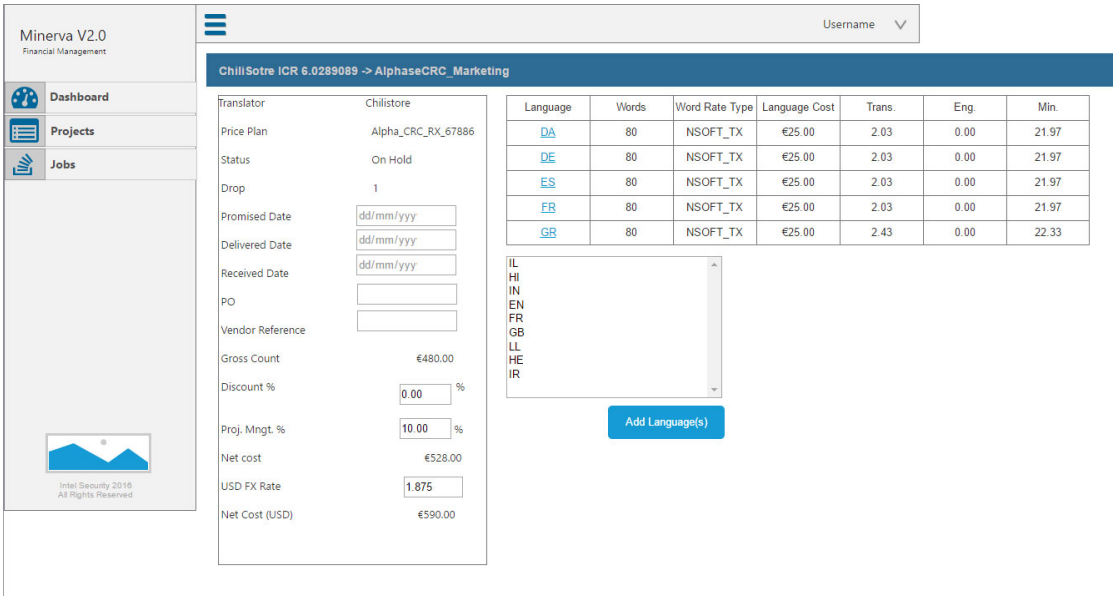


FIGURE A.5: Engineer Job Detail View

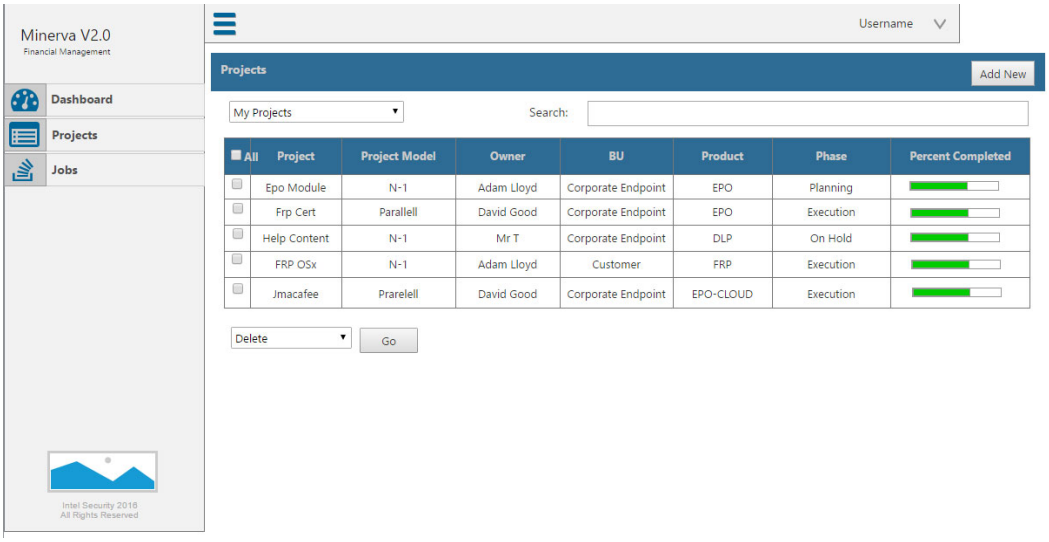


FIGURE A.6: Program Manager Dashboard

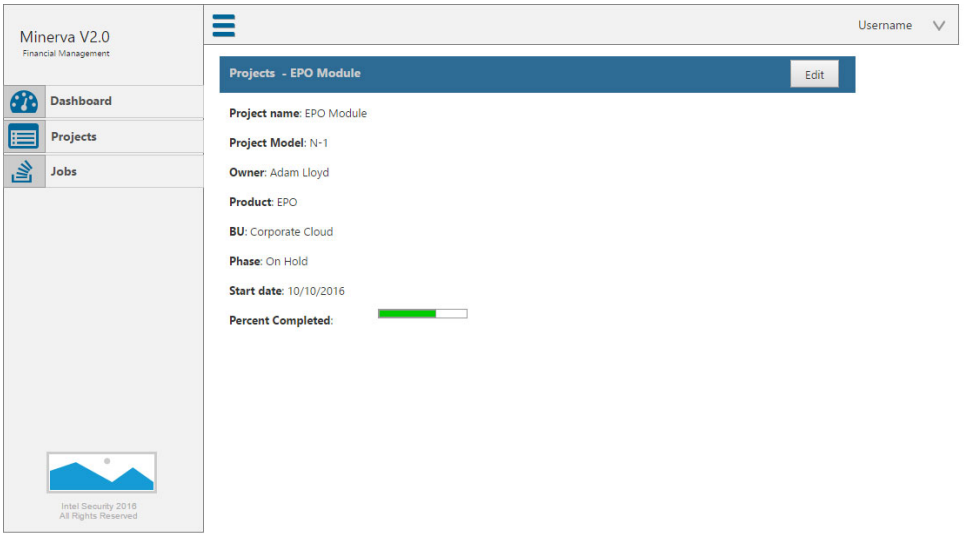


FIGURE A.7: Program Manager Project Detail

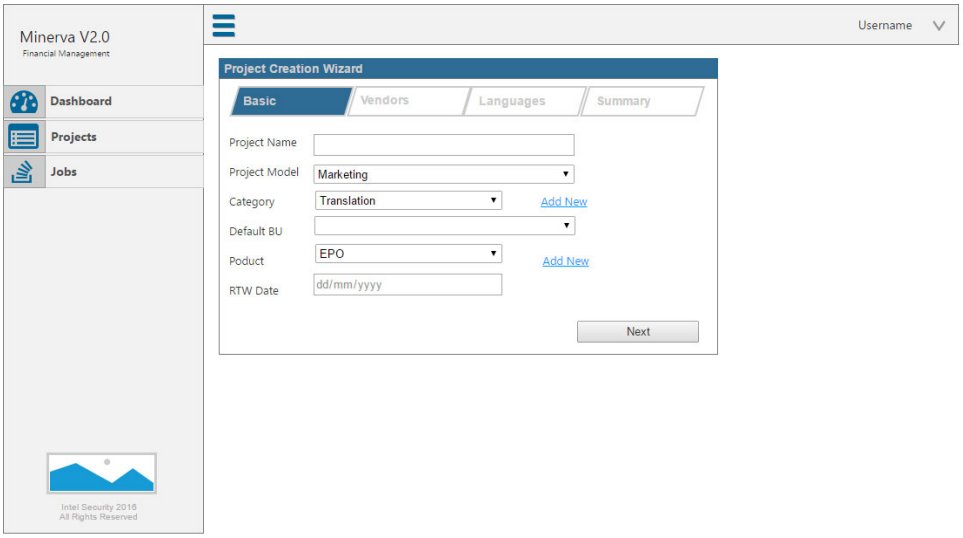


FIGURE A.8: Program Manager Add Project Wizard

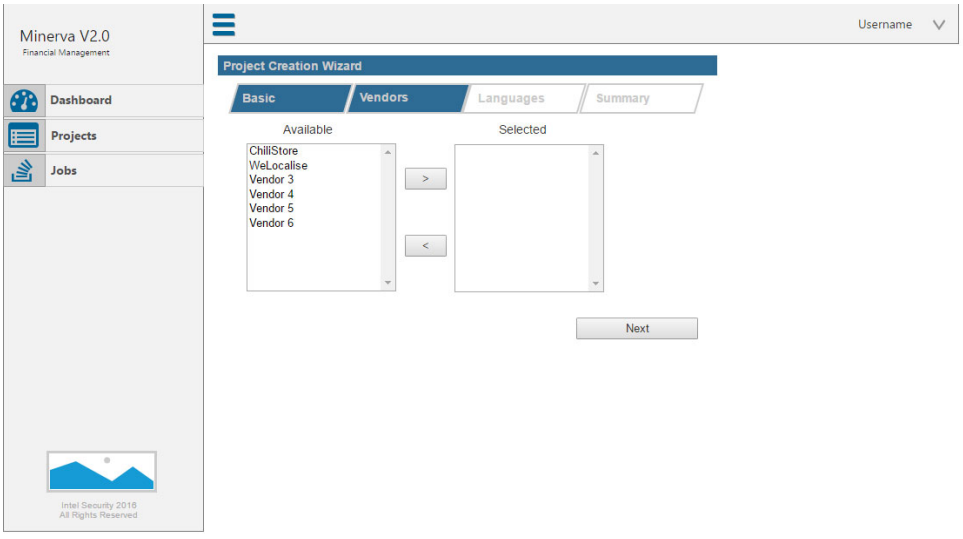


FIGURE A.9: Program Manager Add Project Wizard 2

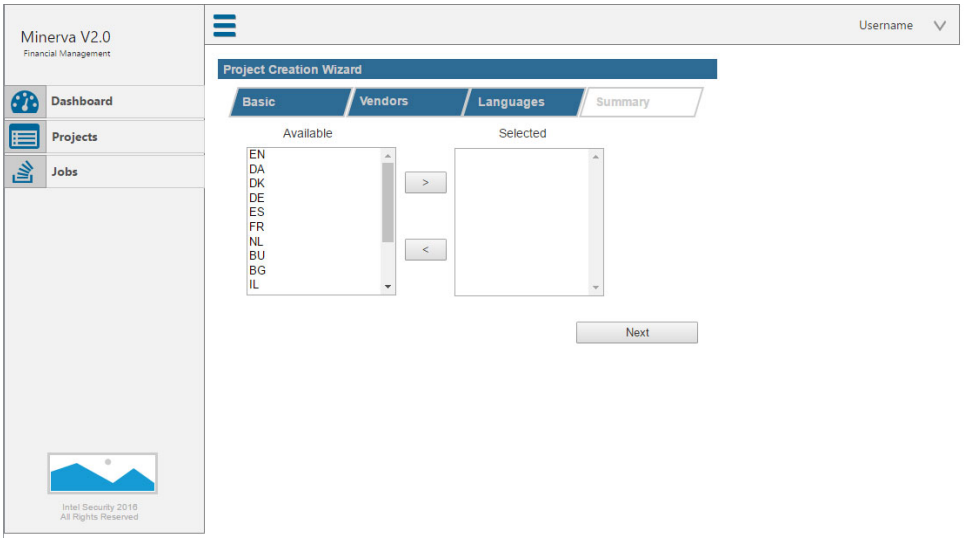


FIGURE A.10: Program Manager Add Project Wizard 3

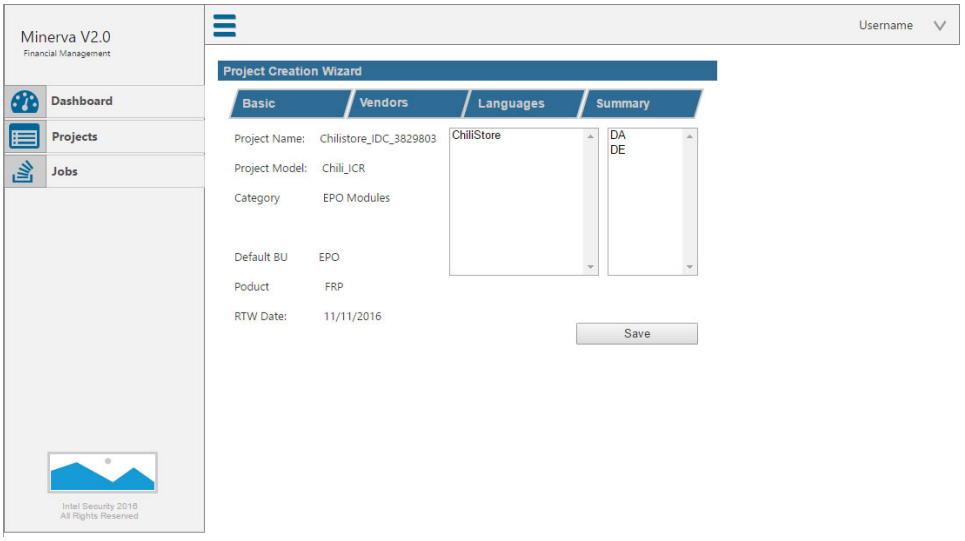


FIGURE A.11: Program Manager Add Project Wizard 4

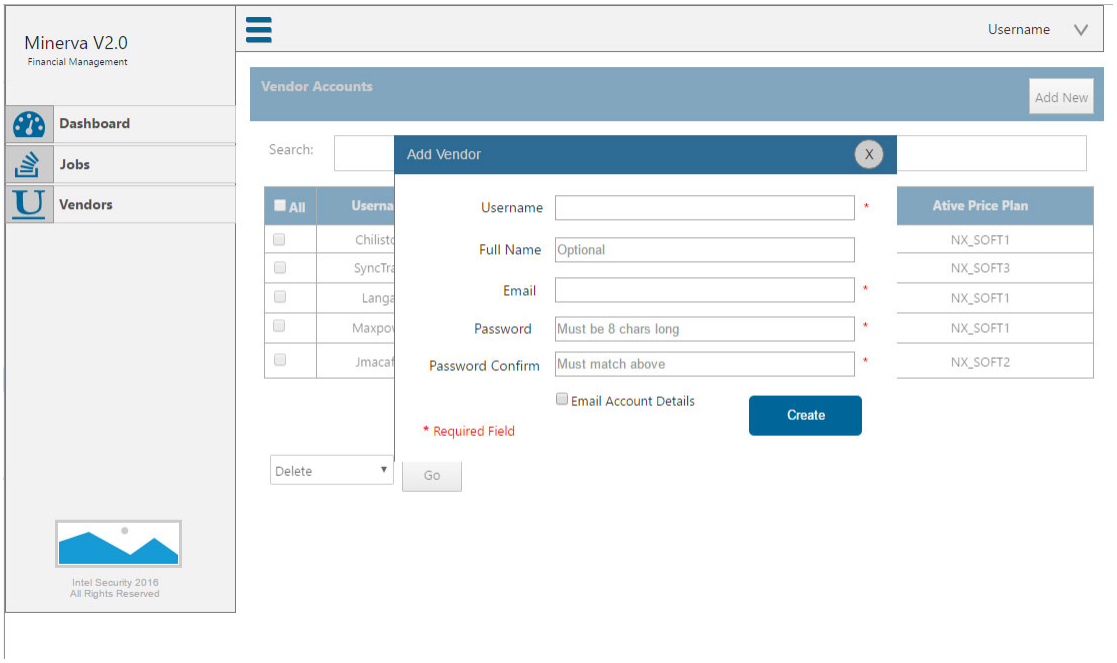


FIGURE A.12: Vendor Manager Add Vendor

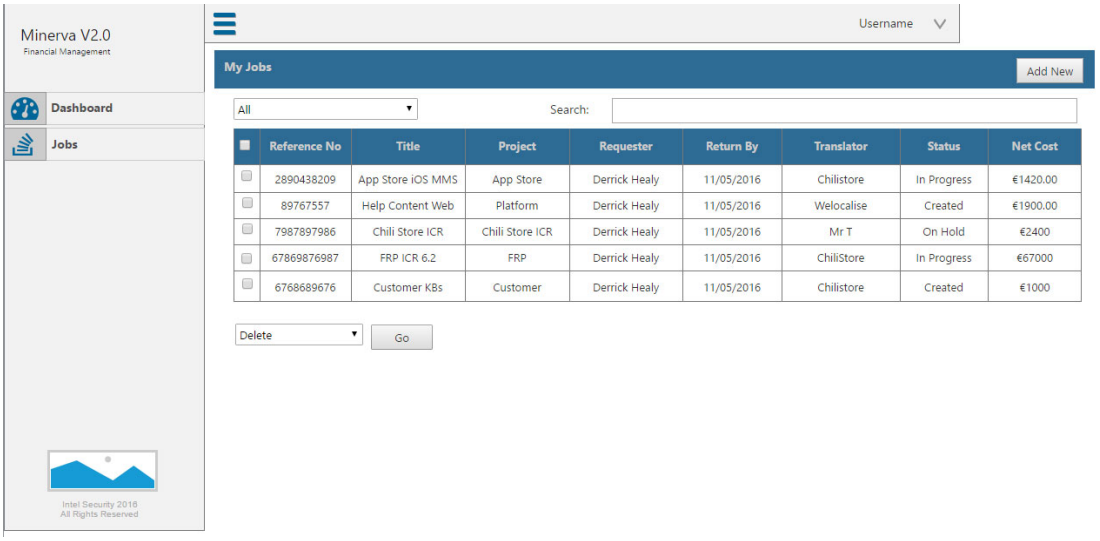


FIGURE A.13: Vendor Dashboard

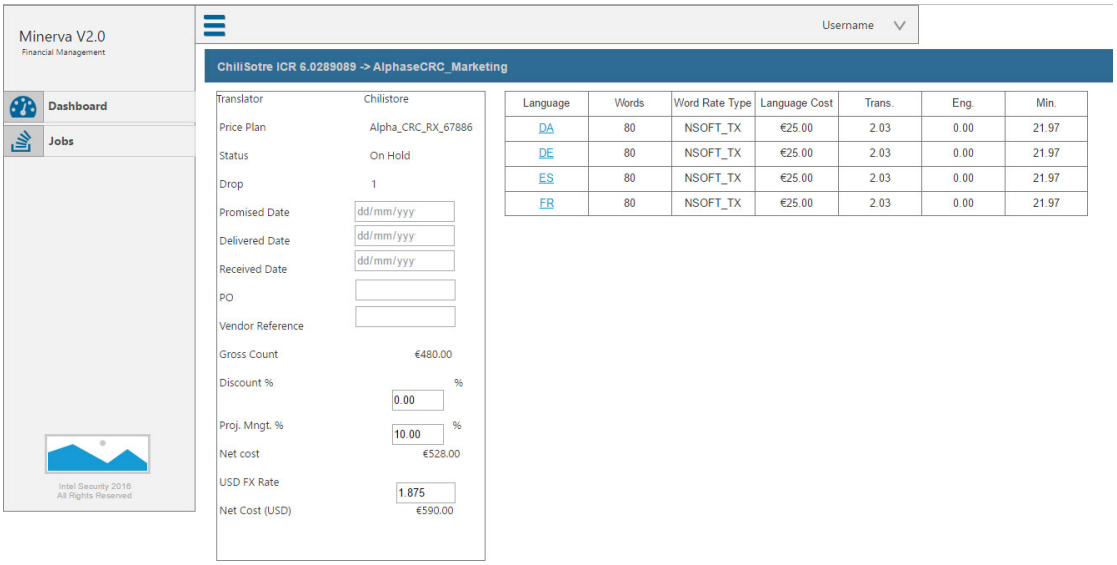


FIGURE A.14: Vendor Job Detail

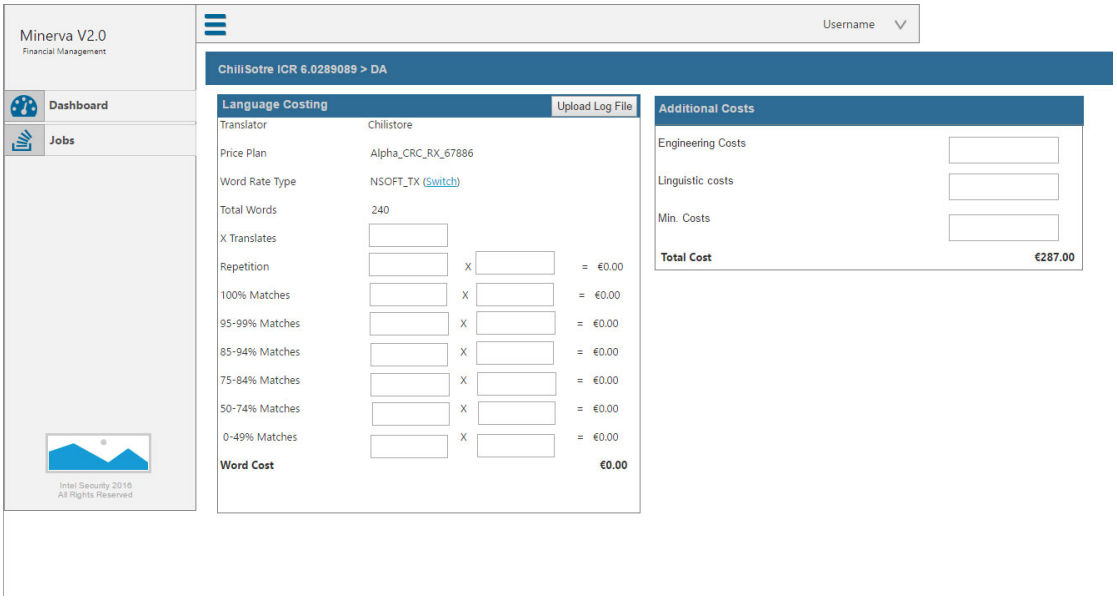


FIGURE A.15: Vendor Job Language Detail

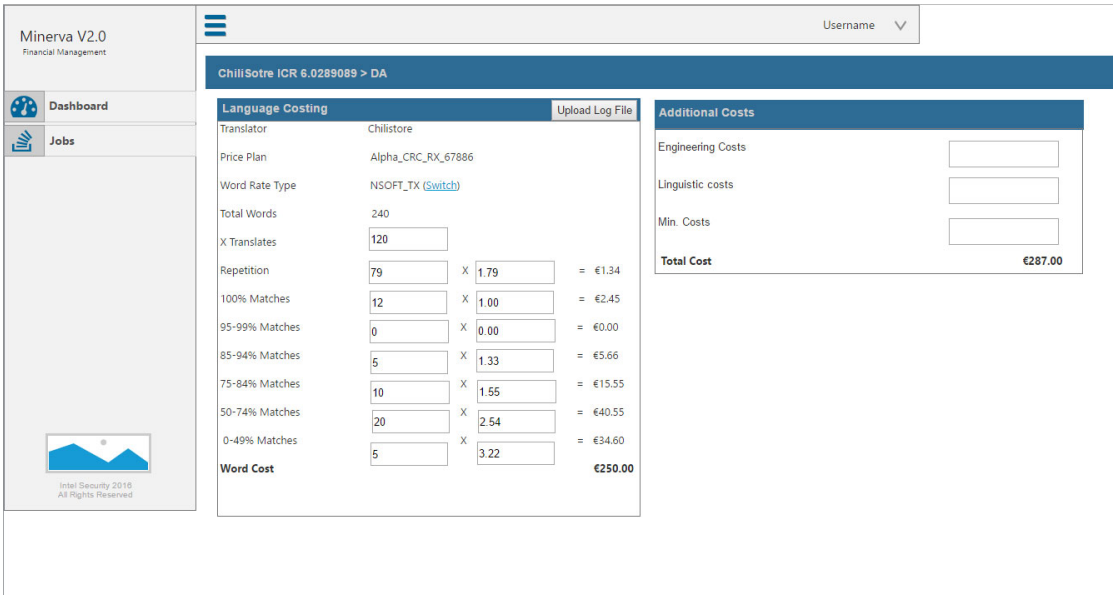


FIGURE A.16: Vendor Job Language Uploaded Log