ASHESI UNIVERSITY COLLEGE

WEST AFRICAN GAS PIPELINE COMPANY (WAPCO) AUTOMATED EXPENSE CLAIM MANAGEMENT SYSTEM

UNDERGRADUATE APPLIED PROJECT

B.Sc. Computer Science

Christian Biassey-Bogart

2016

`

ASHESI UNIVERSITY COLLEGE

West African Gas Pipeline Company (Wapco) Automated Expense Claim Management System

APPLIED PROJECT

Applied project submitted to the Department of

Computer Science,

Ashesi University College

In partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science

Christian Biassey-Bogart

April 2016

`

Declaration Page

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate’s Signature:………………………………………………………………………..

Candidate’s Name:……………………………………………………………………….

Date:…………………………………………………………………………………...

I hereby declare that the preparation and presentation of this applied project were supervised in accordance with the guidelines on supervision of applied project laid down by Ashesi University College.

Supervisor’s Signature:……………………………………………………………………..

Supervisor’s Name:……………………………………………………………...…….

Date:…………………………………………………………………………………...

`

Acknowledgement

I want to take this opportunity to thank my family for their immense support in helping me get this far in my education. I want to also thank my colleagues for helping me keep my head above the water in times of struggle. I finally want to thank my lectures, and most importantly, my project supervisor, for equipping me with the skills and support to successfully complete this project.

`

Abstract

This is the report for the ‘West African Gas Pipeline Company (WAPCo) Automated Expense Claim Management System’ Project. The aim of this project is to properly automate the expense claim workflow within the company. The goal of the system is to make the expense claim process easier and more efficient for employees of WAPCo. Reducing waste of resources such as paper and inconvenient occurrences such as the loss or damage of claim forms.

`

Table of Contents

`

`

`

Table of figures

`

Chapter 1: Introduction

1.1 Background

The West African Gas Pipeline Company limited (WAPCo) is a limited liability company that owns and operates the West African Gas Pipeline (WAGP). The company has its headquarters in Accra, Ghana, with an office in Badagry, Nigeria, and field offices in Cotonou - Benin, Lome - Togo, Tema and Takoradi, both in Ghana. WAPCo is a joint venture between public and private sector companies from Nigeria, Benin, Togo and Ghana. The company's main mandate is to transport natural gas from Nigeria to customers in Benin, Togo and Ghana in a safe, responsible and reliable manner, at prices competitive with other fuel alternatives. WAPCo is owned by Chevron West African Gas Pipeline Ltd (36.9%); Nigerian National Petroleum Corporation (24.9%); Shell Overseas Holdings Limited (17.9%); and Takoradi Power Company Limited (16.3%), Societe Togolaise de Gaz (2%) and Societe BenGaz S.A. (2%) (Company Profile, 2015).

1.2 Problem

Ruth is the supervisor of the treasury department of West African Gas Pipeline Company (WAPCo) and is currently using a combination of paper request forms and excel templates to manage employee expense claims. The current manual process is faced with many errors, such as misplaced expense forms, delayed approvals, difficulty in reporting on claims and waste of paper.

`

1.3 Project Objective

Ruth explained that when an employee wants to claim any expenses, they complete a claims form that includes their personal data and other required information. The form is then sent to finance, to the clerk in charge of budgeting to confirm whether there are enough funds to cater for the claim. If approved by the budget clerk, the form is sent to the line manager who has the authority to approve for the budget code indicated on the claims form. If the manager approves, the form is sent to the director in charge of that budget code for final approval. The approved claims form is then sent to the treasury department for payment. It is important to note that any of the approvers can delegate their authority to one of their colleagues when they are going on leave or will be out of the office and not be able to approve documents. In such a case, the colleague will have the authority to approve requests. The objective of this project is to automate the expense claims and approval process so as to make it faster and more secure.

1.4 Motivation

Travel and expense management is an integral part of any organization. WAPCo being one of such, there are many benefits to automating the Travel and Expense claim management system. For instance, an automated system will be more transparent due to the efficient tracking of transactions and the simplified reimbursement procedure. This system will make the expense claim process faster and more satisfying to employees (Datamatics, 2015). Eliminating the stress and hustle that comes with the current paper and spread sheet system. Not forgetting that an automated travel and expense claim system will also save the company money that is currently lost through human error and misplaced transaction forms. This will go a long way to

`

aid the smooth running of the organization. Policies concerning travel and expense may also be integrated into the system, making implementation smoother and easier (Higgins, 2015).

1.5 Assumptions

For effective use of this system users have to have basic knowledge of the existing expense claim process. The system will therefore be deployed under the assumption that users have a basic understanding of the process and are fairly competent with technology. Not to mention the needed proficiency in reading and typing in English.

1.6 Wish list

This project should largely improve the speed and convenience with which stakeholders perform the expense claim process. The system should seamlessly replace the existing solution of manual paper expense claims without many hiccups or complications. With the standard training that employees receive during the rollout of a new in-house system, integration of the system should be smooth.

1.7 Significance of the project

With the world of business gradually moving into the technological age in Ghana, WAPCo, being the leading transporter of natural gas on the West African coast is keen on evolving its business operations to meet international standards. The Automated Travel and Expense Claim Management System will push further the company's transition into this age. Enabling fast, smooth, efficient and stress free solutions to existing problems.

`

Chapter 2: Literature review

2.1 Introduction

Automated systems date back to the late 18th century with the implementation of the centrifugal governor. A system that regulated the speed of an engine by controlling the flow of fuel (Bennett, 1979). Down the years, a couple of centuries later, automated systems have become increasingly integral to operations in many industries and homes worldwide. These systems perform tasks that were previously done manually. From the use of Automated Teller Machines (ATM) in the banking industry to answering machines in homes, automated systems have made work and life easier and faster for many people around the world.

2.2 Types

2.2.1 Artificial Neural Network

The use of mathematics and computational models to mimic the behavior of biological neurons. Applications of this form of automated systems include speech recognition systems and financial trading systems.

2.2.2 Distributed Control system

The use of a decentralized control system where the controls are connected to other controls by a communication network. These systems are commonly used in industries such as manufacturing and other industries where the production process is continual.

`

2.2.3 Programmable Logic Controllers

These systems operate in real-time, providing results at set intervals. These timeframes are keyed into the system making them suitable for operating machines in manufacturing lines. These systems may also be used to operate other machines such as amusement park rides.

2.2.4 Supervisory Control and Data Acquisition (SCADA)

A large industrial control system that is made up of smaller systems. These include human machine interfaces linked to remote terminal units. Here, sensor signals are converted to data that can be understood by operators. An example is the SCADA system used in natural gas flow control by West African Gas Pipeline Company Ltd.

2.2.5 Human Machine Interface

Also referred to as a user interface, the operation of this system depends on human interaction with the system through the provision of appropriate inputs. The system then provides an output which helps the user achieve his or her goals. A typical example of this kind of system is the ATM used in the banking industry.

(Thomas Publishing Company , 2016)

2.3 Examples of Automated Systems

- Car Manufacturing
- Beverage productions (Example: Coca-Cola)
- Spray Tanning
- Automated Teller Machine (ATM)

`

- Email Systems
- Project management software

2.4 Advantages

There are many reasons why the use of automated systems has become necessary in many industries. A selection of reasons that support the major industrial shift into the age of automated systems are listed below.

2.4.1 Speed

Processes are performed faster than they would be if they are done manually. For example, one of the United Kingdom's most acclaimed bakeries, Geary Bakeries was able to improve quality and increase manufacturing speed by 500% by using the Pulse Width Modulated (PWM) flow control spraying system which ensured that seed toppings were quickly and adequately watered (Spraying Systems Co., 2011).

2.4.2 More accurate

Automated systems are capable of providing improvements in the quality and accuracy of results. Applications are performed precisely and repeatedly without loss of the initial quality. This increased level of reliability may be difficult to achieve by other means. Also, some automated systems are able to run without breaks, allowing processes to be performed non-stop, implying that services rendered are available at all times. An instance of this is the use of an integrated vision system which enabled SC Johnson to package products more precisely for their major overseas clients (machineryautomation.com.au, 2016).

`

2.4.3 Adaptable

Automated systems can be modified to perform processes other than the ones they were originally created to perform. Provided these processes are similar and require a slight change in programming. This may open avenues to even more benefits of using automated systems such as multitasking.

2.4.4 Safety

Automated systems reduce the amount of movement and processes that workers have to perform drastically. Since the movement of documents and other resources may be done electronically. This reduces the risk of accidents that happen as a result of movement required when performing the process manually.

2.4.5 Less Costly

Automated systems reduce organizational costs such as labor costs and the cost of paper that is used in manual processing. Automated systems save the company resource, labor and time costs. For Example, Texi-Pave, a pool coping and paving business shows evidence of this. After switching to automated systems, they were able to reduce running cost significantly. Therefore increasing company profit (machineryautomation.com.au, 2016).

2.5 Disadvantages

Like many systems that have been tasked with performing operations, there are certain factors that may hinder the smooth deployment and use of an automated system. Some of these factors are discussed below.

2.5.1 Worker Resistance

There are many factors that will cause workers to be skeptical about an automated system replacing manual processing in the workplace. Some of these are listed below:

- Employees lose their decision making power.
- Automated systems are seen as an instrument of downsizing.
- Employees feel the constant monitoring of automated systems are an invasion of privacy.
- Automated systems take away the personal interactions between workers when performing day to day activities.

2.5.2 Loss of flexibility

Automated systems are not suitable for some processes that involve workers being malleable and often using their personal judgment. Automated systems often take away this flexibility.

2.5.3 Training

Deploying a new system in an institution requires personnel to receive some amount of training. The institution then has to consider the cost and other factors involved in the training of personnel.

2.6 Conclusion

The choice to create an automated expense claim process of West African Gas Pipeline Company (WAPCo) systems from the facts that have been elaborated upon in this chapter. With examples such as Texi-Pave and SC Johnson, the list of success stories is growing longer and longer. In a quest to make the process faster and less stressful among other benefits, a successful deployment will forward WAPCo’s quest to be a major player in the oil and gas industry. With the help of technology and efficiency in business processing.

`

Chapter 3: Requirement Analysis

3.1 Introduction

3.1.1 Purpose

This system requirements specification (SRS) outlines the needs that the system should cater for, the required functionalities and how best they are going to improve the expense claim process. This SRS is available to all stake holders and parties interested in the system.

3.1.2 Scope

The *Automated Expense Claim Management System* for *WAPCo* will provide a fast and stress free way for employees and the treasury department to deal with the day to day expense claim activities.

3.2 Overall Description

The expense claim process is an integral part of the treasury department of WAPCo. The existing system of paper forms and receipt submissions tends to be tedious, with the back and forth process, and requires a lot of paper management. An automated system will speed up the process allowing time for other important and productive activities.

3.2.1 Product Functions

This will be an online based system. The system will enable employees to perform the activities associated with the expense claim process. The system will also provide an online based approval and workflow, moving request along without the usual cost of transportation.

In this system users will be able to perform essential operations such as making, approving and verifying claims. As well as delegating authority of approval to other employees.

3.2.2 User Classes and Characteristics

The basic user class for this system is the employee population of WAPCo. This system will be used whenever a purchase is made in the name of the company. The user classes of the system are grouped into four categories. The maker of the claim, the head of department, the general manager of the division and the accounts department. Below is a use case diagram that illustrates the different parts that the users of this system will play.

`

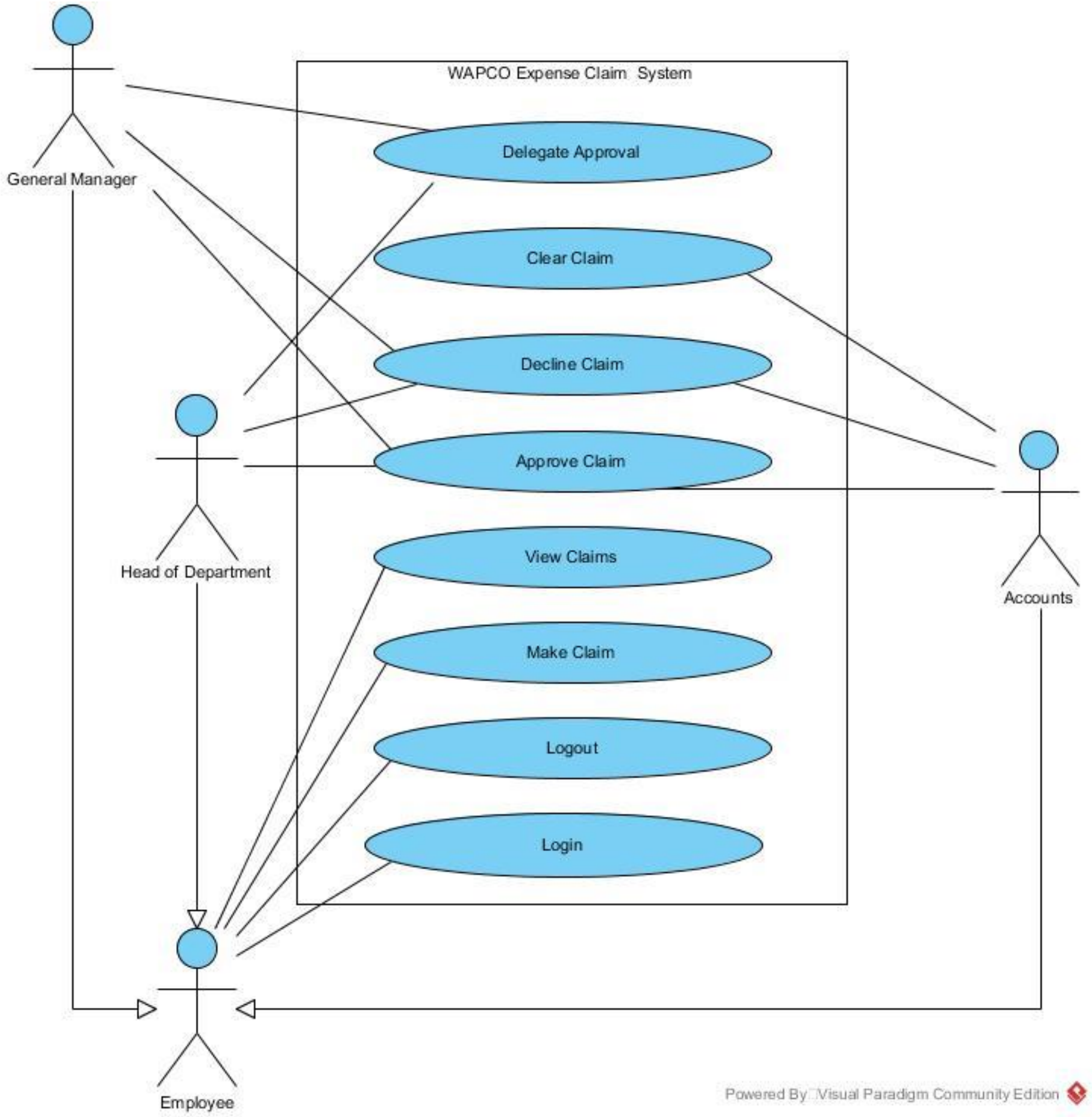


Figure 3-1: System Use-Case Diagram

The diagram above shows the different user classes and the functions they will use to interact with the system. Firstly, the employee will be able to add, edit and cancel claims. Personal claims management is the fundamental purpose of the system in relation to the general employees of West African Gas Pipeline Company. With respect to the different employee roles, the diagram illustrates the additional functions that employees may perform if they belong to the Head of Department or the General Manager user classes. These abilities include

`

approving and declining claims that are made in departments and divisions headed by these employees. A user that belongs to the accounts class will be able to clear claims that have successfully gone through the approval workflow.

3.2.3 Operating Environment

The system will be run on the intranet network of WAPCo which will be accessed by employees on a device with a web browser. The intranet can only be accessed from a company assigned laptop which should be connected to the WAPCo local network.

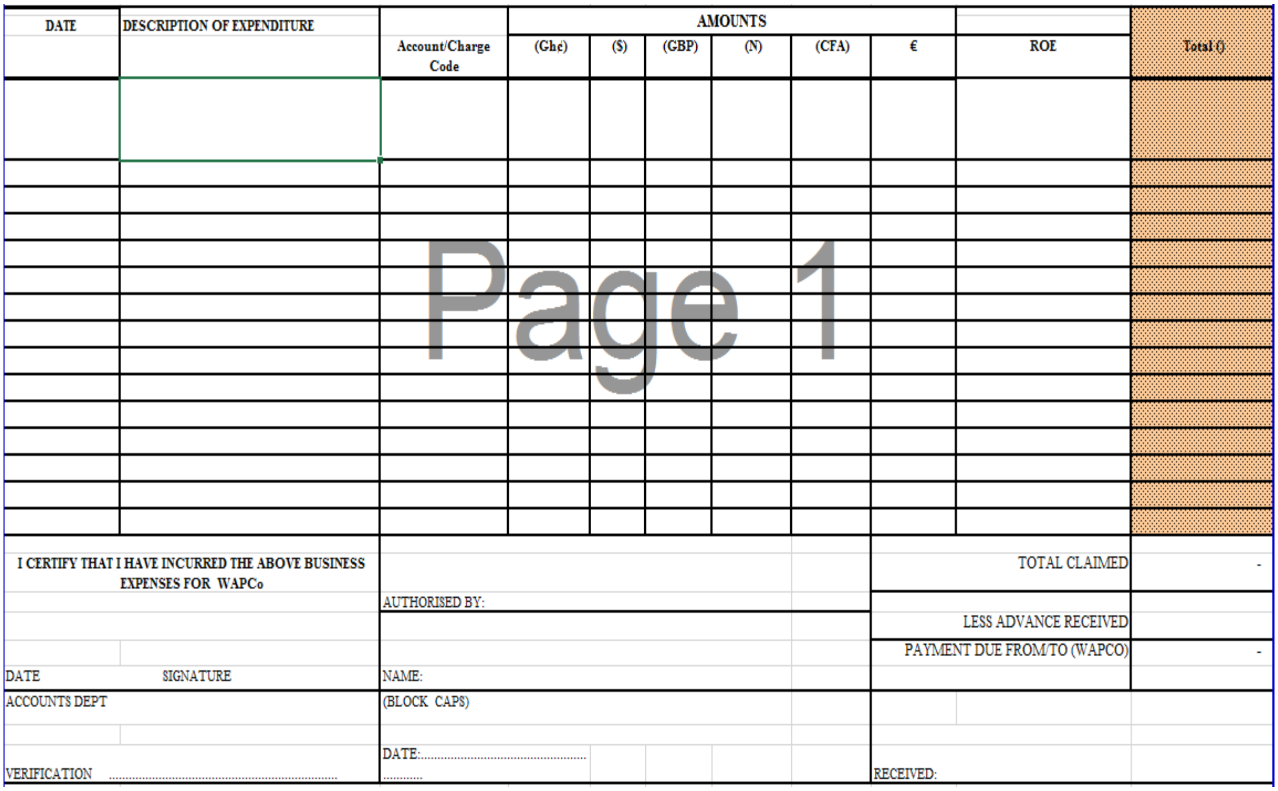
3.2.4 Design and Implementation Constraints

The system will be required to run on windows internet explorer which is the standard web browser installed on company computers, and is therefore the browser that employees will use to access the intranet. The views of the system therefore have to be internet explorer compatible.

3.2.5 Policies

The system will be based on an existing expense claim policy which will guide the functionality of the system. The complete workflow of the expense claim process will be determined by the existing claim policy. Below is the claim form that will govern the design of the form that employees will use in the automated system.

`



DATE	DESCRIPTION OF EXPENDITURE		AMOUNTS							
		Account/Charge Code	(Ghc)	($)	(GBP)	(N)	(CFA)	€	ROE	Total ()

I CERTIFY THAT I HAVE INCURRED THE ABOVE BUSINESS EXPENSES FOR WAPCo	AUTHORISED BY:	TOTAL CLAIMED	-
		LESS ADVANCE RECEIVED	
		PAYMENT DUE FROM/TO (WAPCO)	-
DATE SIGNATURE	NAME:		
ACCOUNTS DEPT	(BLOCK CAPS)		
VERIFICATION	DATE:..	RECEIVED:	

Figure 3-2: Expense claim form

The interface of the system will have to be easy to use and understand since the employees of WAPCo have varying levels of computer literacy.

3.3 Scenarios

To effectively describe how the system will function, the following scenarios have been formulated to illustrate typical instances in which the system will be used.

3.3.1 Scenario 1

Daniel is an I.T Specialist at WAPCo. Daniel bought a hard drive for the I.T department. Daniel logs into the WAPCo expense claim system and chooses the option to add a new claim. He then selects the quantity of items and the currency in which the items was bought in. this

`

information is used to generate a claims form and then fills the form with the product names and prices in their appropriate fields. After filling the form he confirms the claim.

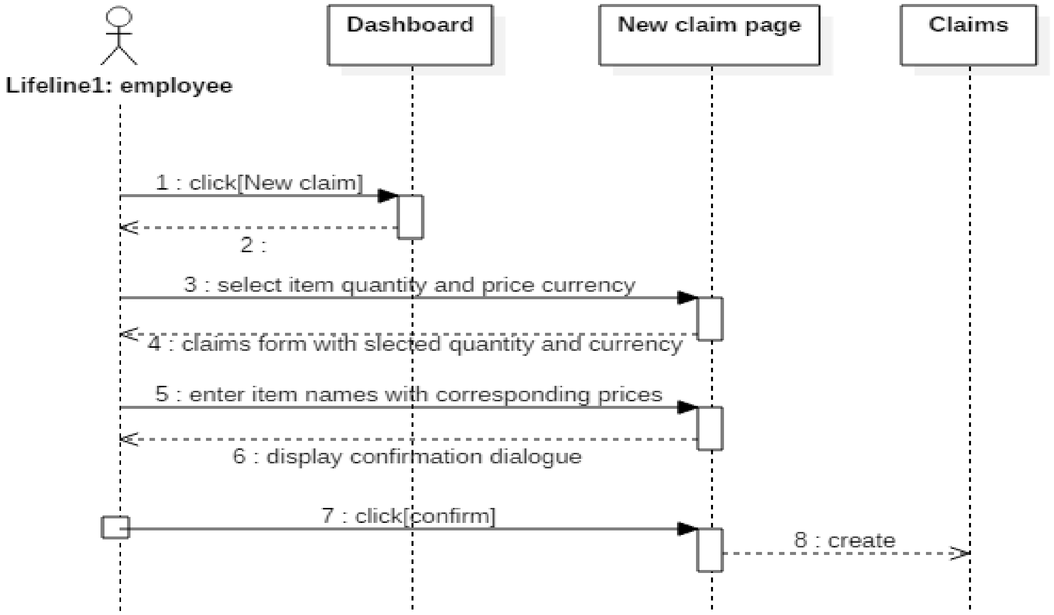

Lifeline1: employee
Dashboard
New claim page
Claims
1 : click[New claim]
2 :
3 : select item quantity and price currency
4 : claims form with slected quantity and currency
5 : enter item names with corresponding prices
6 : display confirmation dialogue
7 : click[confirm]
8 : create

Figure 3-3: Sequence diagram for scenario 1

3.3.2 Scenario 2

Abena is the head of the I.T department at WAPCo. Abena has received an email notifying her that there is a claim she has to approve. She logs into the WAPCo expense claim system and views the pending claims from her department. She then views the claim for approval consideration.

`

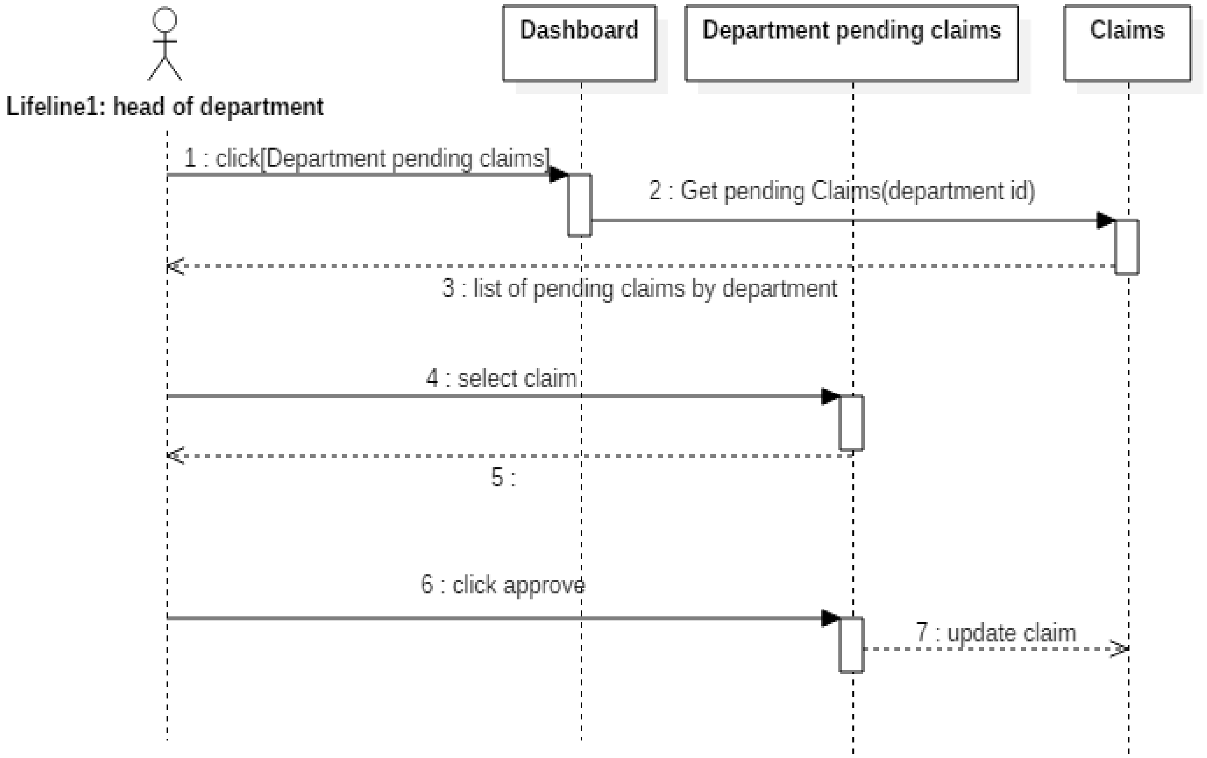

Dashboard
Department pending claims
Claims
Lifeline1: head of department
1 : click[Department pending claims]
2 : Get pending Claims(department id)
3 : list of pending claims by department
4 : select claim
5 :
6 : click approve
7 : update claim

Figure 3-4: Sequence diagram for scenario 2

The above scenarios do not capture the complete workflow from when a claim is added to when it is cleared. To help better understand the complete claims process, the following activity diagram aids in the description of activities regarding claims. These include adding claims, canceling claims, editing claims, approving and declining claims by heads of department and general managers. The process ends when a claim is cleared by accounts.

Below is an activity diagram that illustrates the travel and expense claim process and how some of the various functional requirements contribute to the travel and expense claim process.

`

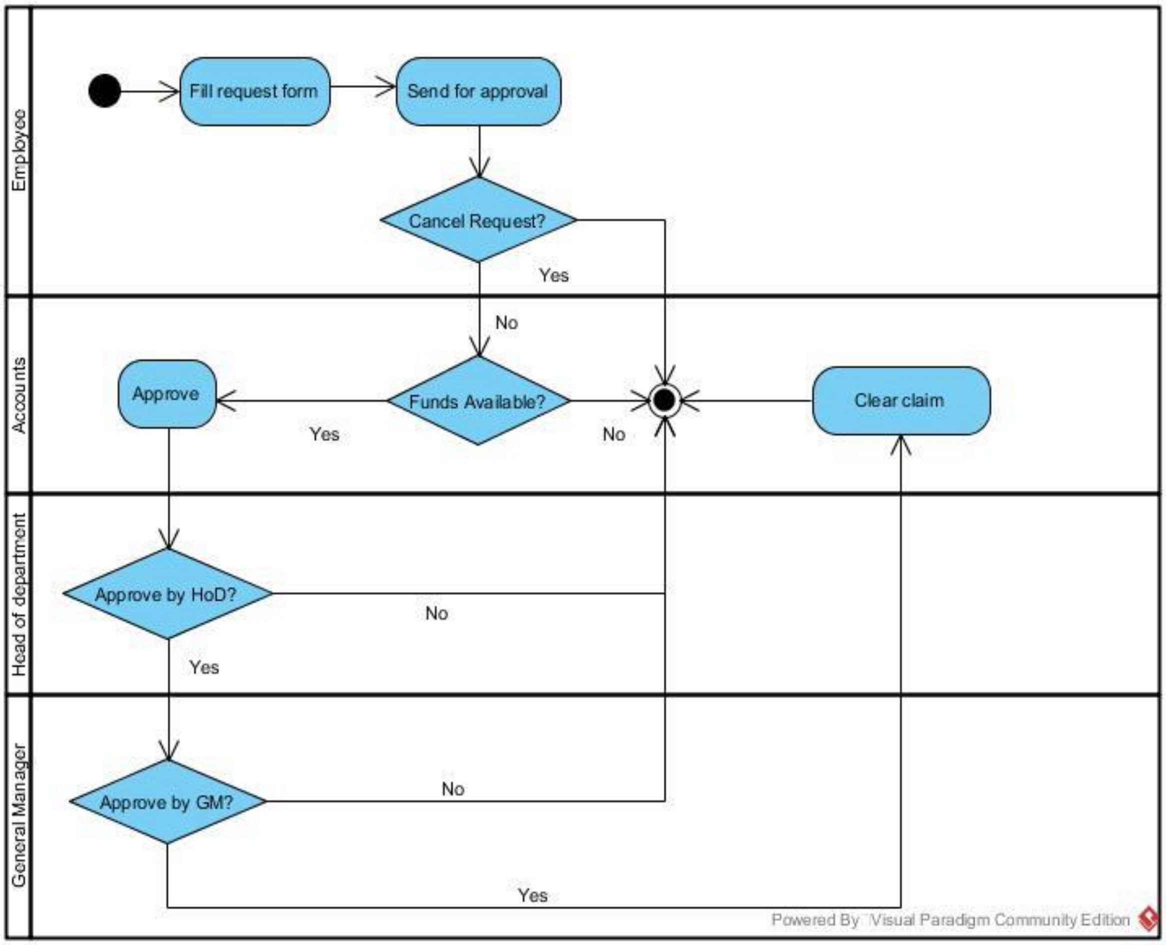


Figure 3-5: System activity diagram

3.4 Specific Requirements

3.4.1 Functional Requirements

3.4.1.1 Online Based Requests: A web based system that will enable employees of WAPCo to request for expense claim online.

a. Add request

b. Cancel request

c. Edit Request

Function	Description	Constraint	Input/output	Testing
Add Request	This function will allow users to add a request to the system	One has to be an employee to add a request	***Input:*** Expense claim information (derived from the expense claim form) ***Output:*** An expense claim with the entered information along with user information is added to the system.	After using the function, the database will be checked to verify if the claim is added.
Cancel Request	This function will allow users to remove their expense claims.	A claim cannot be cancelled if it has already been approved or rejected by the line manager. Only the user who made the claim can cancel it.	***Input:*** The user will select the claim from a list of claims. ***Output:*** The canceled expense claim will be added to a canceled claims pile	After using the function, the database will be checked to verify if the claim has moved to the canceled pile.

Table 3-1: Online based request requirements

3.4.1.2 Online Based Approvals & Workflow: The developed system should incorporate online based approvals based on the workflow outlined by the treasury department.

d. Approve claim

e. Decline claim

Function	Description	Constraint	Input/output	Testing
Approve Request	This function will allow the line manager to approve a request made by an employee.	Only the line manager can approve a request.	***Input:*** The approver will select the claim from a list of claims ***Output:*** The status attribute of the claim will be updated to 'approved'.	After using the function, the database will be checked to verify if the claim status has updated to 'approved'.

Decline Request	This function will allow the line manager to decline a request made by an employee.	Only the line manager can decline a request.	***Input:*** The approver will select the claim from a list of claims ***Output:*** The status attribute of the claim will be updated to ‘declined’.	After using the function, the database will be checked to verify if the claim status has updated to ‘declined’.

Table 3-2: Online based approval requirements

3.4.1.3 <u>Delegation</u>: The system should have the ability to delegate one’s approval authority to other staff members.

f. Delegate request

g. Cancel delegation

Function	**Description**	**Constraint**	**Input/output**	**Testing**
Delegate approvals	This function will allow the line manager to delegate his/her approval authority to another employee.	Only the line manager can delegate approvals.	***Input:*** The user will select the name of the employee that the approvals is from a list of employees in the user’s department. ***Output:*** The approval id of the department is updated to the id of the selected employee.	After using the function, the database will be checked to verify if the department’s approval id has updated to the employees’ id and the delegation is recorded in the delegations table.
Cancel delegation	This function will allow the line manager to cancel an approval delegation.	Only the line manager can cancel an approval delegation.	***Input:*** The user will select the cancel delegation option on their application dashboard.	After using the function, the database will be checked to verify if the approval id of the division or

			Output: The approval id of the division or department will be updated back to the id of the line manager.	department has been updated back to the id of the line manage and the delegation record is removed from the delegation table.

Table 3-3: Delegation requirements

3.4.1.4 <u>Reporting</u>: The reporting required from the system include:

h. Detailed expense report – This keeps track of expenses in the organization.

i. Get report

Function	**Description**	**Constraint**	**Input/output**	**Testing**
Get report	This function will instruct the system to generate a report based on transactions made with the system.	Users will only be able to access reports on claims that were made under their jurisdiction.	***Input:*** The id of the employee who is requesting for the report. ***Output:*** a report will be generated based on the user role.	A printable html page should be created with the information requested.

Table 3-4: Report requirements

3.5 Non-functional Requirements

1. **Auditing**: The developed system can be used to track expense claim activities in the company.
2. **Architecture**: The system developed will follow a three tier architecture – Model tier, Controller tier and Presentation Tier.
3. **Others**: The system will help eliminate the current issues being faced as a result of the manual processing of claims. This includes:

`

a. Misplaced expense forms.

b. Delayed approvals.

c. Difficulty in reporting on claims.

d. Reduce paper usage.

`

Chapter 4: Architecture and Design

4.1 Introduction

The architecture of the system being developed outlines clearly the layout of the system adequately defining the qualities of the system such as security and performance. It serves as a pre-implementation evaluation of feasibility of the system to be developed.

After careful consideration of the various kinds of software architecture, a modified version of the Model View Controller (MVC) is the most suitable architectural pattern to use for this project. MVC pattern enables the user to separate the system into three distinct parts. The model, view and controller.

The model part of the system controls the system data and its interaction with the system. Here the data classes and the system databases are managed and dataflow is controlled.

The view part of the system serves as a presentation platform. This layer determines how the data will be displayed to the user.

The controller part of the system manages the interaction between the system and the user. These interactions are then passed to the view and the model.

The modification in the MVC pattern for this project lies in the use of the controller and the view. In this project, the user interaction and the data representation will be controlled by the view. This will be done using JavaScript as a front end processing framework. The controller will therefore manage the interaction between the view and the model. Translating instructions from JavaScript, which is the front end user interaction processing language, to PHP which is the server side processing language.

`

The relationship between these three components of the system will be explained in more detail in this chapter and the next.

4.2 Advantages of MVC

The MVC pattern holds some characteristics that are essential to how the West African Gas Pipeline Company (WAPCo) expense claim will be developed and operated. These are as follows:

- The MVC is flexible during development. Changes can be made in the particular components without making changes in the others. This enables an agile nature of development allowing the developer to modify the components as needed when unforeseen changes are required to be made during development.
- The pattern also enables uniformity in data representation. It allows the same piece of data to be represented on different views. This is possible since the model and controller components remain constant throughout the system. This will also aid in the accommodation of changes in the user interface of the system. Restricting the scope of change to just the view component. (careerride.com, 2016)
- MVC also allows simultaneous development. With the three components being distinct, the developer is able to develop all of them separately before creating the connections. This improves overall project progress since the developer need not complete the development of one component before starting work on another.

4.3 Disadvantages of MVC

Despite the clear benefits of using the MVC architectural pattern, there are some disadvantages that may hinder the smooth development of the system. These are discussed below:

- Coding of components in the MVC pattern tends to become somewhat complicated as the project grows. The developer has to find the best way to link the components so that the required data is presented in the view and the correct data manipulation actions are carried to the model.
- Frequent updates may be needed if significant changes are made to the model component of the system. This may result in the view or controller components requiring constant updates during development. However, relative to the view, the model component may undergo fewer changes during development.

`

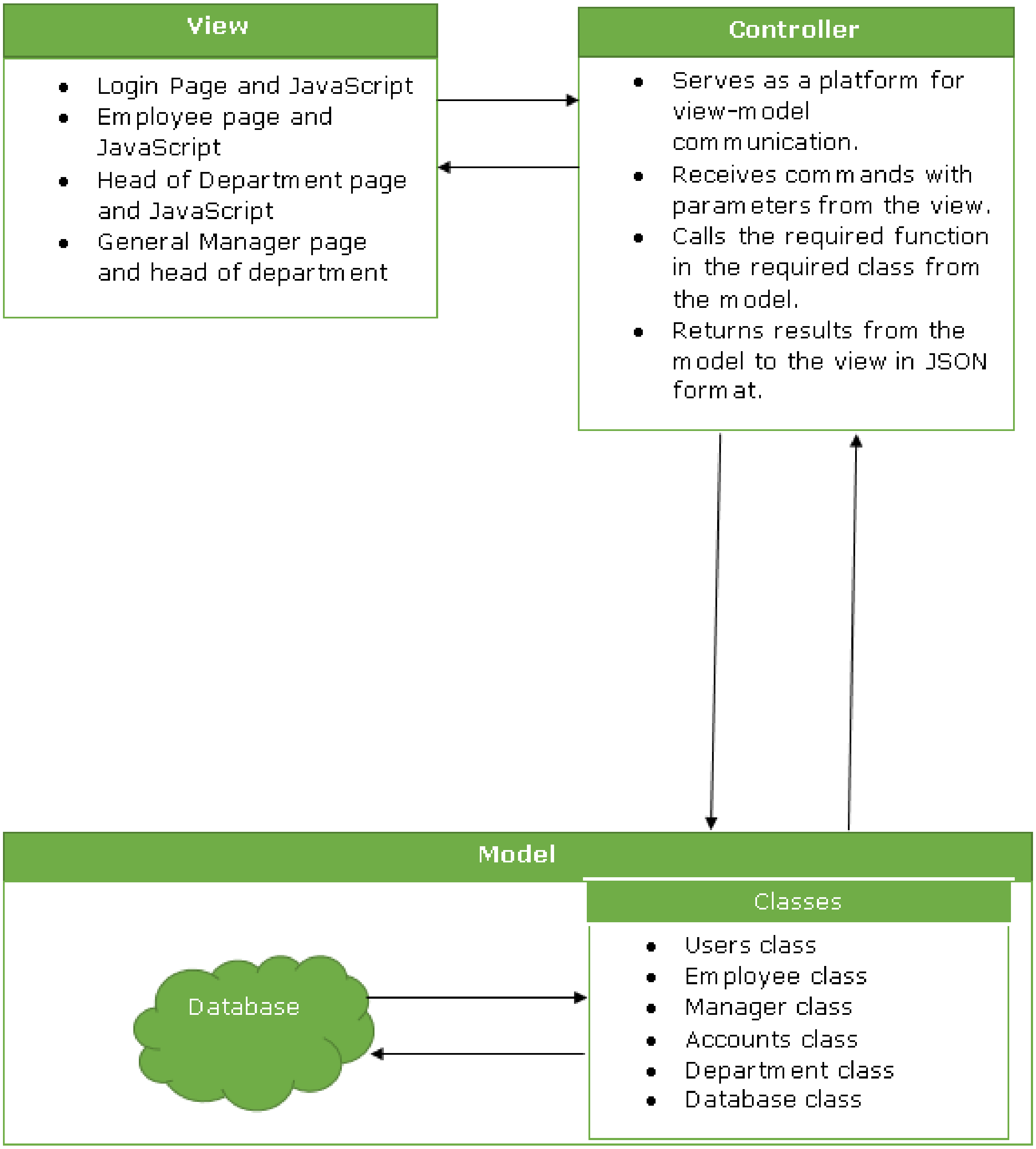

View
Login Page and JavaScript
Employee page and JavaScript
Head of Department page and JavaScript
General Manager page and head of department
Controller
Serves as a platform for view-model communication.
Receives commands with parameters from the view.
Calls the required function in the required class from the model.
Returns results from the model to the view in JSON format.
Model
Database
Classes
Users class
Employee class
Manager class
Accounts class
Department class
Database class

Figure 4-1: System Architecture

`

4.4 Model

The model component of the system is divided into two interconnecting layers. The *database* layer and the *class* layer.

4.4.1 Database Architecture

The database is the fundamental reservoir of information of the system. This section of the chapter will describe the design of the database on which the WAPCo expense claim system is built on.

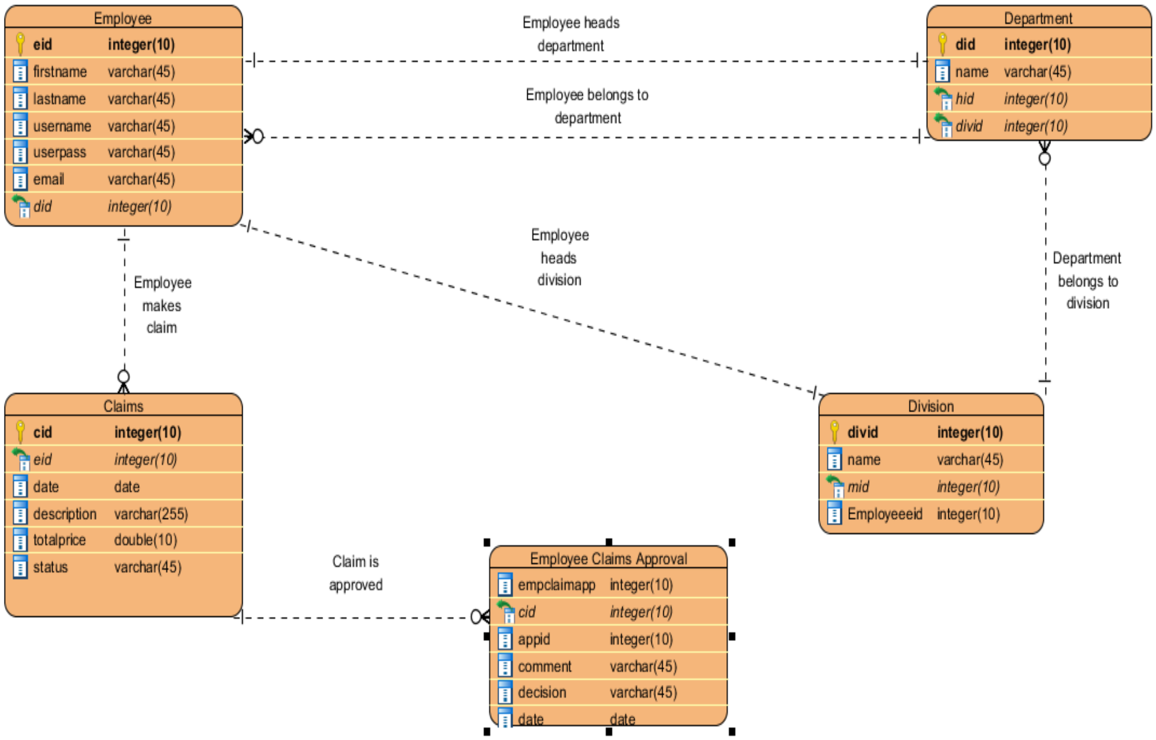


Figure 4-2: ER Diagram for Database Architecture

In the ER diagram above, the relationships between the database tables are clearly illustrated. From the diagram, an employee may make many claims, hence the one-to-many relationships. Between the ‘employee’ relation which holds information about the employees of the company, and the ‘department’ relation which holds information on departments, there is a one-to-many relationship. This implies that a department may contain more than one employee where as an employee may belong to only one department. Also, a ‘division’ is a collection of departments hence the many-to-one relationship between the ‘division’ and ‘department’ entities. The ‘employeeclaimapproval’ entity keeps track of the approvals that take place during the WAPCo expense claim workflow.

The database for the system has been built in a very simple but effective way. The relationships between tables are clear and efficient and retrieving data is fast.

4.4.2 Classes

4.4.2.1 User class

The user class deals with operations concerning user accounts of employees that will use the system.

4.4.2.2 Employee class

The employee class contains methods that deal with an employee’s interaction with the system concerning requirements related to claims.

4.4.2.3 Head class

This class deals with requirements concerning how the head of department works with claims in the system with regards to claims approval.

4.4.2.4 Manager class

The manager class concerns the functions of the system that allows general managers of divisions to work with claims in terms of claims approval.

4.4.2.5 Accounts class

The accounts class contains methods that allow employees of the accounts department perform special functions that the accounts department is responsible for during system workflow. Enabling approval and verification of claims.

4.4.2.6 Database class

The database class deals with all operations that allow the other classes in the application interact with the system database.

4.5 Controller

The controller component of the system regulates the interaction between the view and the model components. The controller receives a commands from the view and translates the command into a PHP method call. The methods are called from the classes in the model.

4.6 View

The view component of the system architecture handles all aspects of the system pertaining to data presentation and user interaction with the system. All interfaces of the system have corresponding scripts that regulate the constant user interaction providing real-time results. This section will be described in two separate parts, the interfaces and the scripts.

`

4.6.1 Interfaces

The interfaces of the system are formed according to the user classes of the users of the system. Namely the administrator, the employee, the head of department, the manager and the workers form the accounts department. With the login interface being the entry into the system.

4.6.2 Scripts

The scripts constitute the processing part of the front-end representation of the WAPCo expense claim system. The scripts that aid the interfaces to function are the login script, the admin script, employee script, head script, the manager script and the accounts. These are directly correlated to the interfaces mentioned above.

`

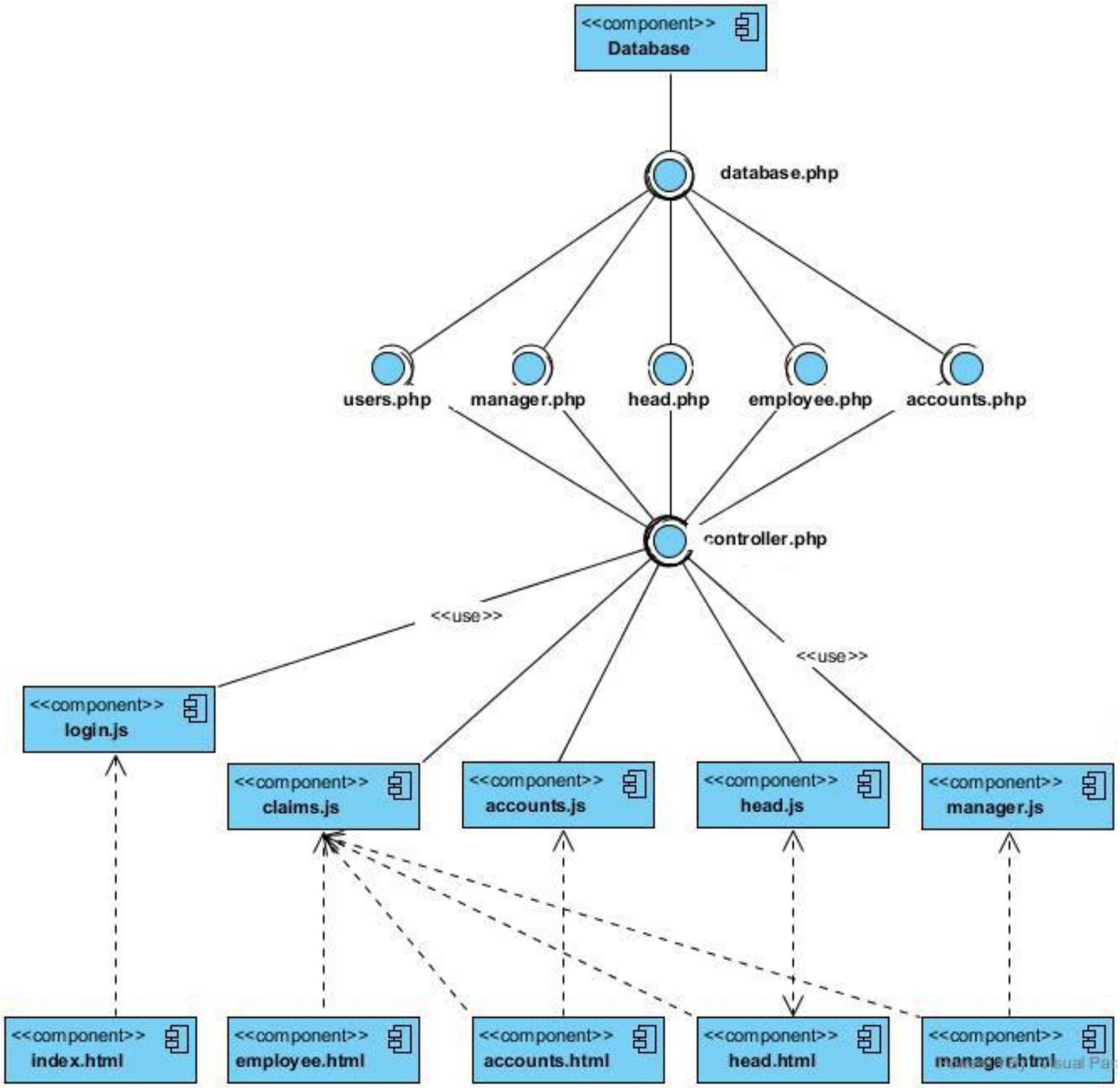


Figure 4-3: System Component Diagram

The figure above illustrates a breakdown of all the system components and how they relate to each other. All interfaces have specialized script files. However the claims.js is used by almost all the interfaces since all users can add claims.

`

Chapter 5: Implementation

5.1 Introduction

This chapter describes the tools that were used to build the WAPCo expense claim system. It also discusses the method that was used to build the system. Furthermore, it describes the functions and interfaces of the various modules.

5.2 Technologies used in building the system

This section will discuss the various technologies that were used in building aspects of the system such as the database, the front end and the server side of the system.

5.2.1 Database Management System (DBMS)

There are many database management systems one may use in setting up a database. These include Oracle, MsSQL, MySQL and PostgreSQL. MySQL will be used in developing this system because it is the most cost effective solution which still provides a high quality database management environment.

5.2.2 Programming languages

Many programming languages for building web applications are in existence. For this project, the programming languages that will be used include PHP, HTML and JavaScript. These three programming languages will be used to develop both the front and server side components of the system. PHP will be used to build the server side component whereas JavaScript, HTML 5 and CSS will be used to build the front end component of the system.

`

5.2.3 Other frameworks

Other frameworks that will be used in developing the system are JQuery, AJAX, PersistJS and bootstrap. JQuery is a library that is used to make JavaScript coding easier. It helps users implement functions with less and more understandable code than was possible before. AJAX improves the user experience by making the client-side element of the system more interactive. AJAX allows developers to create web applications with a reduced number of page reloads and server interactions, making the user experience more streamlined and pleasant. PersistJS is a JavaScript framework that allows developers to use information across pages through storing data as local variables in the browser. Bootstrap is an HTML and CSS library which provides beautification tools that developers can use to create attractive looking user interfaces.

5.3 Modules Developed

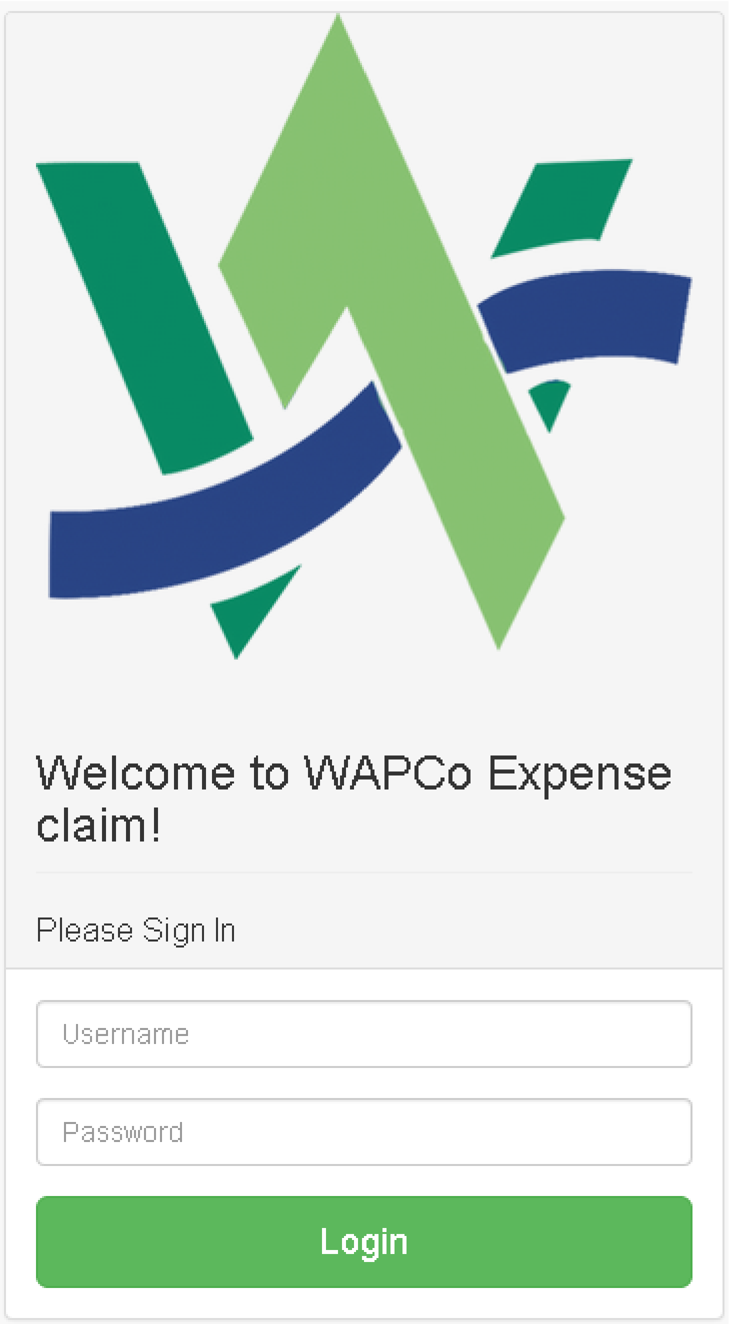
System modules are grouped under the user types. In the section, modules are described using the screenshots and sequence diagrams.

5.3.1 General user

5.3.1.1 Login

This module allows users to log into the system. The user type, which is retrieved from the database when the user logs in, is used to determine the page that the user will be directed to.

`



Welcome to WAPCo Expense claim!

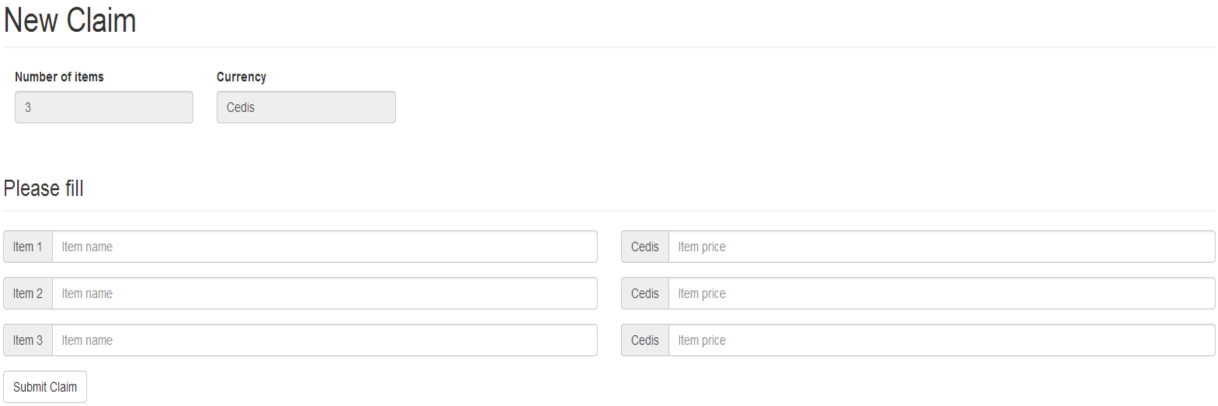
Please Sign In

Figure 5-1: Login module interface

5.3.1.2 <u>New claim module</u>

The new claims module allows all employees to add claims.

`



New Claim

Number of items: 3

Currency: Cedis

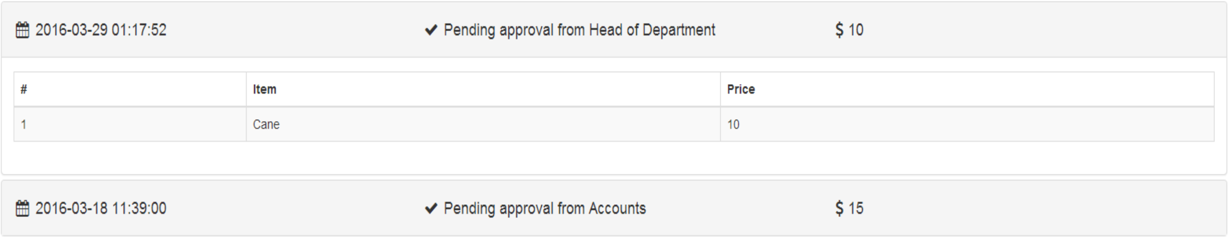
Please fill

Item 1	Item name	Cedis	Item price
Item 2	Item name	Cedis	Item price
Item 3	Item name	Cedis	Item price

Figure 5-2: New claim module interface

5.3.1.3 View my claims module

This module enables users to view claims they have made. These claims display the point at which the claim is in the work flow. For example, in the figure below, it can be seen that the claim is awaiting approval from the head of department.



2016-03-29 01:17:52 ✓ Pending approval from Head of Department $ 10

#	Item	Price
1	Cane	10

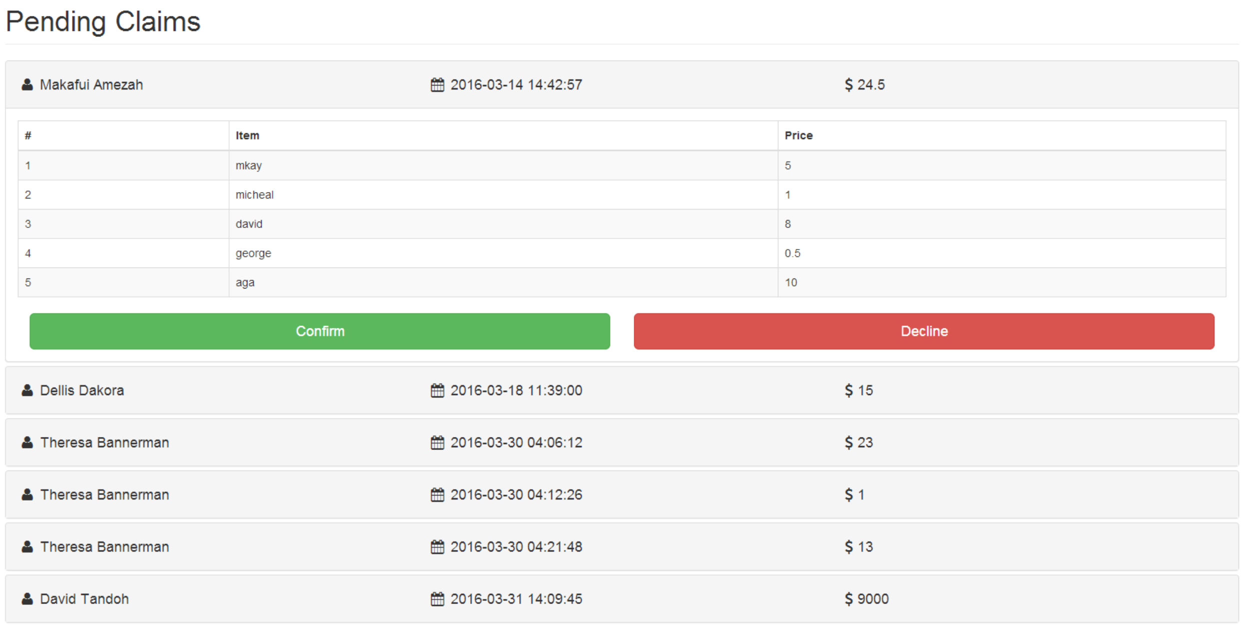
2016-03-18 11:39:00 ✓ Pending approval from Accounts $ 15

Figure 5-3: View claims module

5.3.2 Accounts

5.3.2.1 View pending claims module

This module allows the treasurer to view the list of pending claims.



Pending Claims

Makafui Amezah | 2016-03-14 14:42:57 | $ 24.5

#	Item	Price
1	mkay	5
2	micheal	1
3	david	8
4	george	0.5
5	aga	10

Dellis Dakora	2016-03-18 11:39:00	$ 15
Theresa Bannerman	2016-03-30 04:06:12	$ 23
Theresa Bannerman	2016-03-30 04:12:26	$ 1
Theresa Bannerman	2016-03-30 04:21:48	$ 13
David Tandoh	2016-03-31 14:09:45	$ 9000

Figure 5-4: Pending claims module for Accounts, Head of Department and Manager User classes

5.3.2.2 View approved claims module

This module allows the treasurer to view a list of claims ready for clearance.

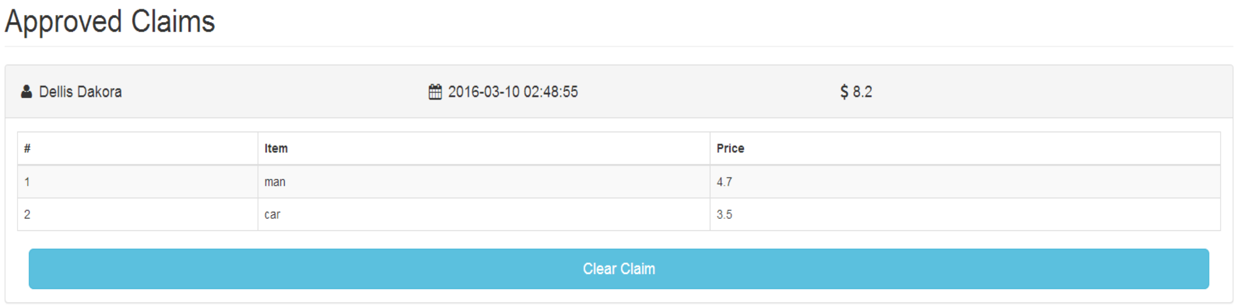
5.3.2.3 Approve claims module

This module allows the treasurer to approve a claim.

5.3.2.4 Clear claims module

This module allows the treasurer to clear a claim. This occurs when the claim has fully gone through the approval workflow. This is the final step in the expense claim process.

`



Approved Claims

Dellis Dakora 2016-03-10 02:48:55 $ 8.2

#	Item	Price
1	man	4.7
2	car	3.5

Figure 5-5: Clear claims module interface

5.3.3 Head of department

5.3.3.1 <u>View my department's pending claims module</u>

This module allows a head of department to view the departments pending claims.

5.3.3.2 <u>Approve my department claims module</u>

This module allows head of department to approve a claim.

5.3.4 General Manager

The manager interface is very similar to that of the head of department and accounts. All three users are involved in claims approval at some point in the workflow.

5.3.4.1 <u>View my division pending claims module</u>

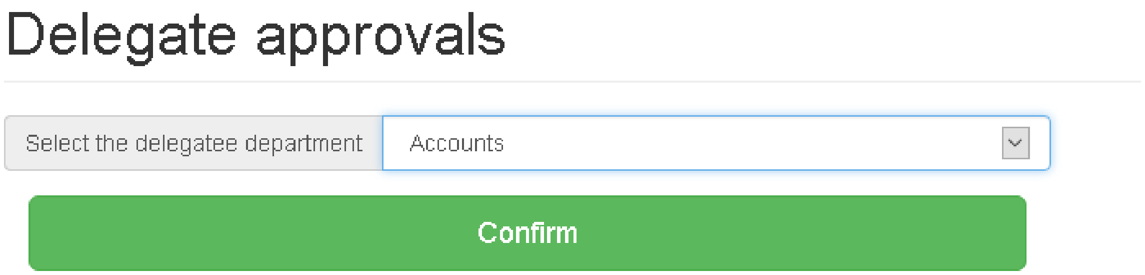
This module enables the general manager to view the pending claims in the division after they have been approved by the head of department.

5.3.4.2 <u>Approve my division claims module</u>

This module allows the general manager to approve claims.

`

5.3.4.3 Delegate Authority


Delegate approvals

Select the delegatee department Accounts

Figure5-6: Delegate approval page

This module allows managers to hand approval authority to another employee in the event of his / her absence.

5.4 Challenges faced during implementation

There were challenges in determining database table relationships for the expense claim system. WAPCo has an existing database. Therefore building a system that is going to run on the WAPCo local network requires a database that will work well with the existing database. Hence it will be a challenge determining the point at which the system database and the company database will connect.

`

Chapter 6: Testing and Results

6.1 Introduction

The mode of testing that will be used to test the system is manual testing. Manual testing was used to test the different components of the system namely the model, view and controller.

6.2 Component Testing

The component testing entails observing whether the database queries and class functions work properly. Testing can be done in one instance since the class function work by calling database queries. Component testing for the six classes will be done by writing PHP code that will call the methods that are being tested.

```
$obj = new department();
$row=$obj ->get_user_by_department(1);
while($row){
print_r($row);
$row=$obj->fetch();
}
```

Figure 6-1: component test code

`

Above is an example of a component test in the department class where an object is created and this object is used to call a method. The results are printed out in a browser when the PHP file is run. This mode of testing is used to test the classes that make up the system.

6.2.1 Component test results

The directory of the class file is called by the browser which in turn run the function within the class file. The results displayed in the browser window are compared to the data in the database.

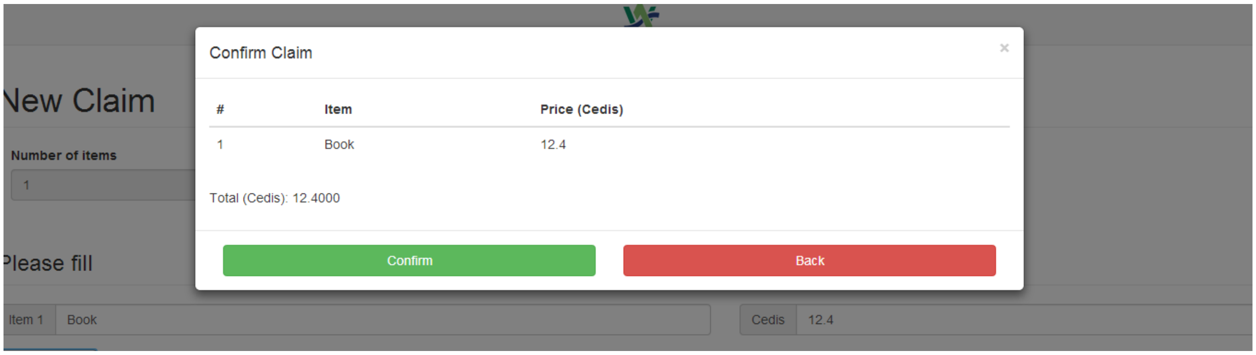
```
1Array ( [eid] => 1 [firstname] => Makafui [lastname] => Amezah [username]
=> makafui.amezah [userpass] => mkay [email] =>
makafui.amezah@ashesi.edu.gh [type] => 0 [did] => 1 ) Array ( [eid] => 5
[firstname] => David [lastname] => Sampah [username] => david.sampah
[userpass] => david [email] => dsampah@ashesi.edu.gh [type] => 1 [did] =>
1 )
```

Figure 6-2: component test results

6.3 System-Level Testing

System testing involves the testing of the combined system, to analyze if the specific requirements are fulfilled on a system wide basis. System test are therefore run at the front-end part of the application. Observing if the required operation was fulfilled at the database level.

`



New Claim

Number of items

1

Please fill

Item 1 Book

Cedis 12.4

Confirm Claim

#	Item	Price (Cedis)
1	Book	12.4

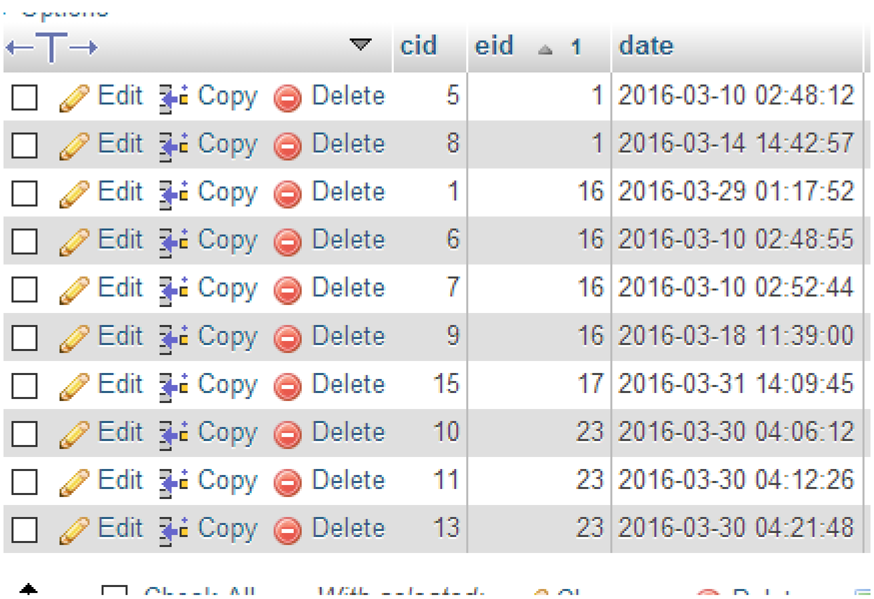
Total (Cedis): 12.4000

Figure 6-3: Add claim system level test

The figure above illustrates a system level test in progress. The test in question is a test to assess if the add claim requirement has been fulfilled.

6.3.1 System level test results

To observe if the requirement has been met, the database is checked to observe if the changes intended by the action at the front-end of the system actually took place.



	cid	eid ▲ 1	date
□ Edit Copy Delete	5	1	2016-03-10 02:48:12
□ Edit Copy Delete	8	1	2016-03-14 14:42:57
□ Edit Copy Delete	1	16	2016-03-29 01:17:52
□ Edit Copy Delete	6	16	2016-03-10 02:48:55
□ Edit Copy Delete	7	16	2016-03-10 02:52:44
□ Edit Copy Delete	9	16	2016-03-18 11:39:00
□ Edit Copy Delete	15	17	2016-03-31 14:09:45
□ Edit Copy Delete	10	23	2016-03-30 04:06:12
□ Edit Copy Delete	11	23	2016-03-30 04:12:26
□ Edit Copy Delete	13	23	2016-03-30 04:21:48

Figure 6-4: Results for system level testing

`

Above is the database claims table which is checked to observe if a claim has been added or updated.

6.4 User Testing

Un-moderated user tests were used to measure the general usability of the system from the perspective of an average user. Students of Ashesi University were used as sample users for this test. From logging in to the addition and approval claims. Users were supposed to explore the application unsupervised. So as to determine how user friendly and effective the system is.

6.4.1 User test results

The system was generally easy to use with a modest but effective interface. Navigation of the system was not difficult and the system performed functions that users required it to.

6.5 General Results

Generally, the system fulfilled the purpose for which it was intended. From the model to the view, testing returned the appropriate results.

`

Chapter 7: Conclusion and Recommendations

7.1 Issues

The complete system works well as a stand-alone application, meeting the agreed requirements. However, integrating the system into the WAPCo framework and modifying the system to work with the existing components may come with difficulties.

Upon discussion with supervisors and colleagues, it was deduced that the current database works well with small quantities of data. However, the database tends to become inefficient when used on a larger scale. Due to the organization and retrieval system of data on claims. This inefficiency was not manifested when testing was performed. Thus these inefficiencies may have to be addressed when the system is being integrated with the WAPCo framework.

7.2 Progress on requirements

The stated requirements have been met in the developed system. Due to policy, the edit claim functionality was removed from the system. This is to ensure that every claim entry has a unique identifying number.

The system will satisfy the project goal of creating a platform where employees and managers can deal with all issues pertaining to expense claims with speed, comfort and convenience. While eliminating the trouble associated with the excessive use of paper.

`

7.3 Recommendations

With the clear benefits that come with converting manual workflow processes such the expense claim process into automated systems. It is recommended that WAPCo continue to create more of these automated systems to replace existing processes, so as to reduce the stress of doing work.

`

References

Thomas Publishing Company . (2016, February 14). *General Types of Automation Systems*. Retrieved from ThomasNet.com: http://www.thomasnet.com/articles/automation-electronics/general-automation-systems

Arthur. (2016, February 14). *Automated Systems Slide*. Retrieved from SlideShare.net:

http://www.slideshare.net/iarthur/automated-systems-slides

Bennett, S. (1979). *A History of Control Engineering 1800-1930.* London: Peter Peregrinus Ltd.

careerride.com. (2016, March 18). *MVC - Benefits of MVC pattern*. Retrieved from careerride.com: http://www.careerride.com/MVC-benefits.aspx

Company Profile. (2015, September 21). Retrieved from www.wagpco.com: http://wagpco.com/index.php?option=com_content&view=article&id=46&Itemid=78&lang=en

Datamatics. (2015, September 21). *Benefits of Travel and Expense Management Services*. Retrieved from www.datamatics.com: http://www.datamatics.com/articles/benefits-travel-and-expense-management-services

DiCaterino, A., Larsen, K., Tang, M.-H., & Wang, W.-L. (1997). *An Introduction to Workflow Management Systems.* Albany, NY 12203: Center for Technology in Government. Retrieved from www.ctg.albany.edu.

Higgins, A. (2015, September 21). *High Tech Companies: Welcome to High-Tech Expense Management*. Retrieved from www.concur.com: https://www.concur.com/blog/en-us/expense-management-software-advantages-for-high-technology-companies_21oct2014_sb

machineryautomation.com.au. (2016, March 17). *Integrated vision system ensures precision product packaging for key overseas client*. Retrieved from www.machineryautomation.com.au: http://www.machineryautomation.com.au/integrated-vision-system-ensures-precision-product-packaging-for-key-overseas-client/

machineryautomation.com.au. (2016, March 17). *Texi-Pave – Total automation solution for paving specialist forecasts impressive return on investment in less than two years.* Retrieved from www.machineryautomation.com.au: http://www.machineryautomation.com.au/texi-pave-total-automation-solution-for-paving-specialist-forecasts-impressive-return-on-investment-in-less-than-two-years/

RobotWorx. (2016, February 14). *Advantages and Disadvantages of Automating with Industrial Robots*. Retrieved from robots.com: https://www.robots.com/blog/viewing/advantages-and-disadvantages-of-automating-with-industrial-robots

`

Spraying Systems Co. (2011). *Bakery Improves Quality and Increases Production Speed Five-Fold with Automated Spray System.* Retrieved from www.spray.com: http://www.spray.com/literature_pdfs/CS132_Bakery_Increases_Quality_Production.pdf