INTEGRATED SALES’ MANAGEMENT WITH MOBILE INTERFACE

SCHOOL OF SCIENCE AND ENGINEERING

CAPSTONE FINAL REPORT

EGR 4402

21\textsuperscript{th} April 2017

Student: \textit{Saâd Laârifi}

Supervised By: \textit{Dr. Nasser Assem}
STUDENT STATEMENT

I, student Saâd Laârifi, am affirming to have applied and followed the code of moral and ethics to the design process and in the selection of the work proposed below. I also assert holding into account the safety of the public to be paramount and have addressed this latter in the presented design wherever may be applicable.

____________________Laârifi Saâd____________________

Capstone Student Saâd Laârifi

Approved by the Supervisor(s)

____________________Dr. N. Assem____________________
Acknowledgements

First, I would like to express my very great appreciation to Professor Nasser Assem for his valuable and constructive suggestions during the planning and development of this capstone project. His willingness to give his time so generously to supervise my work has been very much appreciated.

I would also like to thank the client at Enhanced Technologies, Mme Houda Chakiri, whom is at the head of the company, as she was abundantly helpful and offered invaluable assistance and guidance to make my work within the company as professional as possible.

Furthermore, my sincere thanks and appreciation go to Dr. Yassine Salih Alj, who is the capstone coordinator here at Al Akhawayn University. He helped make the process of working on the final capstone project as fluent and simple as possible.

Finally, my deepest gratitude go to my dear friend Hicham El Alami, whom worked with me the first parts of the project as a joint Capstone with the same company and client. He assisted me during multiple times of need and contributed in many ways in the success of the project.
Contents

STUDENT STATEMENT ........................................................................................................... 2
Acknowledgements ................................................................................................................... 3
Contents ................................................................................................................................... 4
Abstract .................................................................................................................................. 6
Introduction .............................................................................................................................. 6
STEEPLE Analysis .................................................................................................................... 8
Socio-Cultural ............................................................................................................................ 8
Technological ............................................................................................................................ 8
Economic ..................................................................................................................................... 8
Environmental .......................................................................................................................... 8
Political ...................................................................................................................................... 9
Legal .......................................................................................................................................... 9
Ethical ........................................................................................................................................ 9
Requirement Specifications ....................................................................................................... 9
Functional Requirements .......................................................................................................... 9
  Introduction ............................................................................................................................ 9
  Intended Reading Audience ................................................................................................... 10
  Project Scope ........................................................................................................................ 10
  Product Perspective .............................................................................................................. 10
  User Classes and Characteristics .......................................................................................... 13
  Operating Environment ........................................................................................................ 14
  Design and Implementation Constraints .............................................................................. 15
  User Documentation ............................................................................................................ 15
Non-Functional Requirements .................................................................................................. 15
  External Interface Requirements .......................................................................................... 15
  Performance Requirements ................................................................................................... 16
  Safety Requirements ............................................................................................................ 16
  Security Requirements ......................................................................................................... 16
  Software Quality Requirements ........................................................................................... 16
Feasibility Study ....................................................................................................................... 17
  Economic ............................................................................................................................... 17
  Operational ............................................................................................................................ 17
  Technical ............................................................................................................................... 17
  Resources .............................................................................................................................. 18
Methodology ............................................................................................................................ 18
Software Architecture ............................................................................................................. 20
Design ....................................................................................................................................... 21
  Database ............................................................................................................................... 21
  Use Case Diagram ............................................................................................................... 25
Implementation of The Solution ............................................................................................... 30
  Language used ..................................................................................................................... 26
  Paradigm used ...................................................................................................................... 26
Abstract

This document is a summary of the work accomplished during the realization of the capstone project within Al Akhawayn University with the cooperation of Enhanced technologies. My work consisted of designing and creating a backend for enhanced technologies to manage their sales as well as a mobile application to be able to view the necessary information.

The project is a joint-capstone with student Hicham El Alami and with the supervision of Dr. Nasser Assem as well as the help and support of the client Enhanced Technologies.

The project started with the development of a database using MySQL as a database management system, and the development of the according back office website to manage the database that was coded in CakePHP 3. The last and most important part of the project was based on developing a mobile application for the targeted users of Enhanced technologies to view needed and necessary data that would help in the efficient growth of the company.
Introduction

The capstone project is a final project that must be presented within the last year of any American system bachelor degree. It is a comprehensive project that encapsulates everything that has been learned during the curriculum. This project consisted of the development of a mobile application from scratch for an IT company. This type of project regroups most, if not all the components of computer science learned; from database design, website development, software engineering, to mobile development.

Enhanced Technologies is an IT company involved in the digitalization of civil status documents. It is also involved in the sales of other technological components. The digitalization of those documents also includes the distribution of tools and technologies to make the solutions accessible to their clients. Their type of work requires important documents to be managed closely to keep track of their actions and to avoid any mistakes or loss of data. The purpose of this project is to provide a computerized solution to their selling and purchasing process. Today, Enhanced technologies has paper based records. To keep up with today’s trends and keep up with the competition, the company must move to a more sophisticated solution for their business.

The software and application that I am implementing in this capstone provides a way to store their bills, quotes, purchases order, and so on in everything that relates to the management of their sales through a well-designed backend and back office that would be accessible from a smartphone mobile application.
STEEPLE Analysis

Socio-Cultural
Our society is becoming more and more digital after each decade. As our application is going to be a mobile one, it shows that it follows the flow of society in focusing to make everything accessible to the smallest of devices anywhere. Today’s generation follows the trend of smartphones where everything that is needed in one’s everyday life is contained in his or her mobile phone, including work material. This project will allow the people at Enhanced Technologies to follow today’s lifestyle, while at the same time allowing efficient and well-organized work.

Technological
Technological advances have greatly changed the way businesses operate. Nowadays, technological progress created a society which expects instant results. This means that the rate at which information is exchanged between suppliers, clients, and companies has greatly increased. This application will enable Enhanced Technologies to keep up with the technological trend and growth that we are perceiving today. Furthermore, it will enable the company to have a faster exchange of information which will benefit them as they will be able to react faster to changes.

Economic
Since this solution is based on open source tools, languages, and software, it means that it will be a free one. This means that the company will have more revenues for other purposes. This could also mean that it will influence the purchasing power of Enhanced technologies, it might even reduce the prices of the solutions that Enhanced Technologies offers because they have less expenses. This latter presents a great asset against the competition in the same field.

Environmental
While this project doesn’t have much to do with the environment or the ecology, we can say that it will transform the company from a paper based one to a digitized one. In that context, this also
means that there will be a significant decrease in the use of paper in the company, which is always in the benefit of the environment.

**Political**
No direct political consequences or analysis that can be drawn from the deployment of our solution.

**Legal**
Our solution does not infract in any way the legal rules imposed for the usage of software or technological tools. All regulatory and law determinants within this field have and will be respected without any modification needing to be made. Also, the results of this solution will not be affected by legal matters. Furthermore, as the Enhanced Technologies proposes solutions for the digitization of government paperwork, we would say that the information that our client utilizes is very critical and confidential, and should not be disclosed to the general public. In this sense, the solution will also allow to keep track of the actions of the employees within the system, which is an important addition to the business to contain leaks and enforce accountability.

**Ethical**
The code of moral and ethics has been respected to come up and develop this solution. It complies will all morality and integrity behaviors, and acts for the best interest of the company, its employees, and society as a whole.

**Requirement Specifications**

**Functional Requirements**

*Introduction*
The product is a software system, database back office plus mobile application, to automate the sales management process of the company. The product includes the management of the users, employees, sales management, and inventory management. The functional requirements described below cover both the back-office management and the mobile application.
Intended Reading Audience
This document is intended for the eyes of the supervisor, the second reader, as well as the rest of the software engineering school, including faculty, current, and prospecting students as it will be stored in the SSE Capstone repository.

Project Scope
This project consists of a backend office which allows interactions with the database to view, add, edit, and delete data following the business logic of the company. It also consists of a mobile application that will allow access to some critical data when on the move. The business logic deals with the deployment of multiple IT solution, which requires a capable system to keep track of the progress with the clients, especially in the invoicing sections. The goal of this project is to facilitate, automate, and accelerate the transmission and storage of vital information. This increase in the rate of exchange of information will enable the client to react faster and hence benefit positively from this product.

Product Perspective
This product or software could be considered as a self-contained product or as a replacement to the existing paper based system. This SRS describes two fragments of the developed product:

1. Back-office Website: This is the backbone of the product as this is where the business logic is defined and implemented. This interface allows users to add articles, new bills, manage existing bills, manage employees, and so on.
2. Mobile Application: This is a follow-on product to the back office, as it will fetch data that added to the database by the back-office, and will display it to the user through a mobile application. The data can’t be added or deleted through the application as this latter will only allow operations such as display, search, sort, and update.

Back-Office Website

- Authentication

  Deals with the different users that have access to the system and their level of access. This part controls the users as it allows to create, edit, and delete
accounts and control their privileges and limit the actions they can perform and the information they can view.

**Sales Management**
Deals with the orders received from the clients to provide IT solutions. It includes management of quotes, delivery forms, bills, Invoicing, and clients’ management.

**Purchases Management**
Deals with the interactions that the company has with its suppliers. It includes management of quote requests, Purchase orders, bills, payments, and management of suppliers.

**Inventory Management**
Deals with the stock of articles that the company has. While this component does allow addition, modification, and deletion of articles, it is mostly going to be used to view the articles available and the corresponding quantity on hand. The modification that will affect the quantity available of the articles will mostly be applied after confirming the delivery of an article - whether to a client or by a supplier – from other components such as Sales and Purchases.

**Administrative Component**
Deals mainly with employee management as well as some minor data such as adding or modifying bank’s information or modifying the header that gets displayed in the forms generated by the company.

*Mobile Application*

**Authentication**
Deals with user access to grant rights to view specific data depending on the rank of the user or the employee logging in – These rights and privileges are set in the Authentication component of the Back-office website.

**Inventory Viewing**
Allows the mobile application user to view information related to the articles available in the stock, to sort them based on different criteria, or to search for a specific article.
• **Employees Viewing**

  Allows the user to view information related to the current employees and to search for a specific employee. This component is mainly used for contact purposes, instead of having to store the contact information of every employee in the company in one’s mobile phone.

• **Sales Information Viewing**

  This component will allow users to view pending orders, requested quotes, recent completed invoices, and even information pertinent to the company’s clients. Once again, this is relevant to have the client’s contact information handy whenever needed.

• **Purchases Information Viewing**

  This component will allow users to view pending quotes that the company made to its suppliers, purchasing orders, invoices pending to be paid to the suppliers, recent payments made, and suppliers related information. The same need for contact information applies here.

Below is a diagram representing the major components of the system and how they relate and interact with each other. This is only a high-level representation of the system. The details will be discussed in further sections of this report. Note that the abbreviation **CRUD** in the diagram stands for Create, Read, Update, and Delete Privileges.
User Classes and Characteristics

The product developed within the framework of this project offers different usages depending on the user. The users are distinguished and differentiated based on the technical expertise, security and privileges levels, and/or experience. The following user classes are defined in the system:

- **Admin**: This is the entity that can add, modify, or delete user accounts. This is the technical personal that is going to be responsible for the good functioning of the system. While this user can have access to the core of the system and can perform the totality of
the actions that other users have access to, the only actions that are pertinent to it are the management of the users and the stability of the system.

- **Employee with Full Access**: This entity can perform all the actions that the system offers except for user management. It inherits from the employee with moderate access and has additional functionalities such as management of clients, management of suppliers, and management of moderate access and lower privileges employees.

- **Employee with moderate Access**: The moderate access employee acts on the core business logic of the system. This entity has access to the Sales, Purchase, and Inventory component as it can perform CRUD operations on element pertaining to those three to make use of the advantages of the application.

- **Employee with Viewing Access**: This entity can’t make any type of modification to the database. This mean that it can only view data that it has access to, such as viewing bills, viewing articles, querying the database to return the fields from tables, and so on.

**Operating Environment**

As the system is both web based and mobile based, it requires a certain environment to run properly. The core of the system is firstly based on a MySQL database. Thus, the system first requires a server that runs on a stable version of MySQL. The system was tested under the v5.7 version of this latter. The database server can run on an independent server or in the same deployment server.

The system also requires a webserver to host and deploy the back-office management website. It could work under any HTTP server such as Apache. The website can then be utilized using any browser without any additional configuration. The system can be configured to allow access either only from the local network of the company or from any internet access point, depending on the client’s preferences – This configuration can be made in the hosting Webserver. The database server and/or the Webserver can be configured to work in any server with any operating system.

The mobile application was developed using PHP and Android studio, which means it can be made into an application in a smartphone, or can be opened as a web service or website in any
other platform that doesn’t support android as an operating system. For that purpose, the application can also be deployed in a server to be accessed as a website from any phone if the hardware available is not compatible with the form of PHP that was used.

**Design and Implementation Constraints**

The constraints imposed on this project before the beginning of the development was to only use open source tools as to follow the client’s practices. Hardware limitations included developing the database using MySQL since the company is already running a database in a server using MySQL, and it was preferable to create a new database in the existing DBMS instead of having to install another database server and lower the performance of the server.

MySQL Workbench was recommended as the client has experience using that tool. The language and technologies to use were not specified as long as they were free for professional usage and were compatible with most platforms in case of a migration of hardware, architecture, or operating system.

**User Documentation**

User manual will be provided to the client for directions and understanding of how to use the system. The manual will include description of how to perform tasks relevant to the business logic of the company. Documentation will also be provided to troubleshoot known problems that may occur when using the system.

**Non-Functional Requirements**

**External Interface Requirements**

**User Interfaces**

No Graphical user interface standards have been imposed by the client: The design of the screen layout, the buttons, the keyboard shortcuts and so on were left to the software engineer to decide upon. However, screen layouts constraints include compatibility with any screen size and type, meaning that it should be set dynamically by the graphical user interface of our system depending on the information it gets from hardware it is used on.
Hardware Interfaces
The software layer of the product was developed to interact with the database which can be stored in any type of hardware. It is preferable that it would be stored in a local server, but the access to the database can be modified to adapt any kind of hardware that reaches minimum configuration.

Communications Interfaces
The system is compatible with all web browsers, but requires the HTTP communication standard.

Performance Requirements
While there are no specific performance requirements with specific numbers, the system should be fast enough to not hinder the flow of business as it could be critical in the growth of the company. In that sense, the product should be able to load and perform actions and operations in a speed that is acceptable and does not require significant and noticeable latency.

Safety Requirements
Actions that modify the database in a significant way should not be allowed except under close supervision of the person responsible of the data. Thus, access privileges should be implemented properly and access to the database should be permanently and closely supervised. In addition to that, daily and constant backups of the database should be made as any sudden loss or damage to the information stored in the database could be fatal to the business.

Security Requirements
User Accounts should be kept secure and private. If the system is configured to enable access even from outside the local network, leaking user accounts with high privileges could disclose pertinent data that is supposedly confidential to the public eye.

Software Quality Requirements
Other quality characteristics for the product that are important both for the client, developer, and IT personnel responsible for the maintainability of the system are:

- **Availability:** The system should be always available for usage except for specific times for backups of the database that should be announced in advance to the personnel.
• **Correctness**: The core business logic that the system assumes is right should be correct as the clients want to avoid any fallacies in the logic.

• **Maintainability**: The software should be easily maintainable, meaning that dependencies to external and indirect components can be updated and configured with ease.

• **Reliability**: The system should be dependable and consistent if the business is to be based on it.

• **Usability**: The system should be easy to use and easy to learn for current and prospecting employees of the company.

**Feasibility Study**

**Economic**
Since the solution has to be developed using open source tools, there will be no incurred cost from deploying such a system. Thus, the Cost/Benefit ratio shows a huge advantage for this product. The company already owns servers that host the software solutions they market; one of those servers can host the webserver and the database. The client also owns the necessary desktop computers and smartphones to utilize the solution. Therefore, from an economic point of view, the project is feasible.

**Operational**
Considerable changes will be brought to the flow of operations that the business process of the company is today based on. Once the system is deployed, basic computer usage skills will be required by the employees to use it. The product is developed to meet certain usability standards, meaning that it should be easy to use and easy to learn. Thus, little training should be required to get used to the information system. From an operational point of view, the project is feasible.

**Technical**
The problematic at hand does require substantial technical requirements. As it consists of a basic Enterprise resource planning system, the level of technology that is needed for the development of the product matches the technology available to the engineers for the development. As
mentioned many times earlier, the system is to be developed using open source tools, meaning that the developers have access to these tools. The programming languages can be chosen by the developers themselves to use what they are comfortable with. As for the hardware resources, such a project can be developed and tested on a single computer before being deployed across servers and stations. From a technical point of view, the project is feasible.

**Resources**

While the project seems feasible from the economic, operational, and technical aspects, the human resources and time available still have to be considered. While the first half of this project is joint between two students, it seems that the time given is close-fitting compared to what should be produced. To increase the chances of success and decrease the odds of failing to deliver at the right time, the right methodology has to be used, and be used properly, in order to be efficient. For that purpose, agile software development will be applied to this project as described in the next section, mainly rapid application development.

**Methodology**

Seeing the extent of this project, and the short duration under which it has to be implemented – a little over 3 months, choosing agile software development as a methodology or process to try and achieve the set goals is the best course of action to undertake. Agile software development describes a set of principles for software development under which requirements and solutions progress through the combined effort of both the software engineers and the client. This methodology comes with incredible benefits such as flexibility of the software requirements as it is delivered incrementally to the client and modifications can be made on any step of the software engineering process. Customer involvement is also a crucial factor in these conditions, especially that Enhanced Technologies is located within the Techno park of Al Akhawayn university, which means that the customer is nearby and can constantly assess the progress of the project.
Furthermore, agile development induces regular face to face conversation with the client, which is the best form of communication in software engineering to make sure that the requirements specifications and needs are well understood. In brief, agile development is based on the following principles, that are also very helpful for the success of the capstone project or any other software project for that matter:

- Focusing on the needs of the client and communication as this latter is prioritized over the selection of processes and tools.
- Correctness and reliability of the software is prioritized over comprehensive documentation.
- Flexible development is prioritized over sticking with a firm plan set at the beginning of the project.

At the same time, considering an incremental model or waterfall model approaches were out of the discussion due to the nature of the project in hand. The method had to be fast, effective, efficient, flexible, and compliant to the deadlines of submission. A simple contrast between the two methodologies is shown in the following diagram:

![Figure 2: Traditional Methods VS RAD](image-url)
The system was implemented using CakePHP which uses the Model-View-Controller Paradigm. As shown in the figure, the database communicates with the REST API which defines a set of functions that developers can perform requests on and receive answers to via protocols such as GET and POST. This same API communicates with other APIs. The REST API is part of the webserver that hosts the system, but the database can be hosted on a different server.

The controller components of the system defines the functions or the business logic that can be applied or fetched by the view component to display the data.
The user connects to web application using web browsers from their computers. The web browsers communicate with the system through HTTP protocols such as Apache. This webserver communicates with the database which can be stored in another server. The user can also use a smartphone with the mobile application installed, which communicates directly with the database server.

**Design**

**Database**

The first step in the design process in software engineering is to analyze the data. When analyzed, the data of a business details the core logic and helps the engineers to come up with the components needed in a certain system. For that reason, gathering data is a critical step in the correct understanding of the needs.

After heavy work on the database, the following Entity Relationship Diagram was generated:
For the sake of the report, I will only explain in detail the tables that are related to the Sales component as the same logic is mirrored in the Purchases Component:

- **Company**

This table will store data related to the company using the system. The attributes include the name of the company, contact information, bank information, and other redundant information that only need to be set once and that will be used in other forms. The purpose of this table is to fetch the data whenever it’s needed in some form instead of having to type it every time.

- **Employee**

This table will store data relevant to the employees of the company. It is important to store such data to keep track of the workers and to have contact information handy. In addition to that, we want to control the flow of information that employees have access to and we want to supervise the actions they undertake.

- **Product**

This table contains the set of articles that the company will market or owns for its own usage. Its attributes are the ID of the product, the label of the product, the category of the product, the price per unit, and the quantity on hand - available in the stock. The quantities of these products will be affected by actions such as finalizing a sale or a purchase.

- **Category**

This table only contains categories for the products. This is used as an enumerated data type to store the categories that can be modified later on. This category is used as an attribute to add a product.

- **Client_ste**

This table contains the information and attributes related to the clients of the company. Among this information are contact information and bank information that will be fetched when generating a form directed to a specific client.
• **Province & Commune & City**

All these tables are there to store static information to enforce consistency of data. These can be considered as attributes to the client table but they are implemented as different tables to enforce certain inputs because the geographic location of a client is an essential information in the business that the client is engaged in.

• **Client_phone**

This table will include a series of phone numbers with the ID of the corresponding client that it belongs to. This was not stored as an attribute to the client_ste table to allow clients to have multiple phone numbers where they can be contacted.

• **Sale_quote**

This table represents the quote that the company generates for the client. It includes a reference to the client that it is generated for, the date of creation and expiration, the id of the employee who created it, and so on. This table in itself does not contain the articles and their quantity, but its line table refers to it for each different article.

• **Sale_quote_line**

This table represents the lines for a specific quote. This is necessary because a quote may contain multiple articles, therefore this bridge table is needed. Its attributes consist of the reference or the id of the quote that it refers to, the id of the product, the quantity, and the price at which it will be sold.

• **Purchase_order_client**

This table represents the purchase order that the client makes to the company after agreeing on a certain quote. It contains the same attributes as the sale quote and refers to a certain quote.

• **Purchase_order_client_line**

These lines refer to lines in the Sale_quote_line table, which are then represented in the Purchase_order_client table.
• **Sale_delivery_form**

After the purchasing order is made by the client, a sale delivery form is generated from that latter and is stored in this table. Once the delivery is made, the corresponding items that were declared in the form are modified in the articles table to automate the calculation of the quantity on hand of the articles.

• **Sale_bill**

This is the corresponding bill that matches the items in the delivery form. The bill is invoiced to a specific client and refers to the purchase order made by that same client. When the bill status is declared to have been paid, the transaction is complete and the Bill is not pending payment anymore.

**Use Case Diagram**
The diagram above shows a representation of the users involved in the system as well as the actions they can perform depending on their access privileges. CRUD operations in the diagram stand for Create – Read – Update – Delete operations. The description of the actors is as follow:

- **Admin**: Creates, updates, and deletes user accounts.
- **Manager**: Can perform all the actions of the employees, in addition to actions involving CRUD operations on employees, suppliers, and clients.
- **Employee**: Acts on the business logic of the system
- **Mobile user**: Reads, updates, and sorts through multiple tables such as the employees one, supplier, client, bills.

**Language used**

The language that was used for the development of this solution is CakePHP 3 which is a framework of PHP that allows rapid software development. It offers rapid building of prototypes using scaffolding and code generation. Very little configuration is needed and the environment is very simple and light to set up. This web development framework can be used starting the fifth version of PHP.

Furthermore, this framework presents important advantages as it follows the trend of development using the MVC paradigm to guide the clean and proper development of an application or system.

**Paradigm used**

The MVC paradigm is an important technique in today’s web development networks. It is a software architectural pattern that differentiates between different component of a software to allow easier development, simpler and consistent modification, and most importantly professional code readability. The diagram below shows how the three components interact with each other within the scope of this MVC pattern:
Because the components are separated into different files, we can avoid reinventing the wheel by being able to reuse code and components of the software that were already developed to reproduce scenarios and views that are somehow similar.

The Model Layer

Represents the part of the application or software that incorporates the core business logic. It is responsible for fetching data from the database and acting on it to convert it into meaningful representation of the concepts of the application.

As an example, this is the model of the Bill table which is an important and critical part of the business logic of the client. As perceived from the code snippet, it sets the fields of such a table and defines the relationships and foreign keys with other tables in a conventional way to automate the generation of further code.
The View Layer

Is responsible of rendering the modeled data that is used in the controller and the model for it to be presentable in the web application. The view components analyses the business logic set in the model and the functions included in the controller to be able to generate a website page that follows the core logic of the business, meaning that it will generate the necessary fields to perform CRUD operations, while also taking into accounts relationships with other pages.

![Figure 11 Add View for Bills](image)

This is the page that was generated by the view from the business logic gathered from other components. This page is responsible for adding new bills, we can see from the snippet that it prompts the user to input the reference of the bills, the client that it is directed to, and also allows him to add multiple articles to the bill and their quantity to generate a complete business form that can be used professionally.
The Controller Layer

This is the component that interacts with the users of the system. It takes as input requests from the users and tries to provide the answer back with the proper output after analyzing the model and view previously created. The controller determines what the user wants and decides to which component it should delegate the task to, before returning the answer to the user.

```php
public function view($id = null)
{
    $bill = $this->Bills->get($id, [
        'contain' => ['Companies', 'Articles']
    ]);

    $this->set('bill', $bill);
    $this->set('_serialize', ['bill']);
}

/** *
 * Add method *
 * @return Cake\Network\Response|null Redirects on successful add, renders view otherwise *
 */
public function add()
{
    $bill = $this->Bills->newEntity();
    if ($this->request->is('post')) {
        $bill = $this->Bills->patchEntity($bill, $this->request->getData());
        if ($this->Bills->save($bill)) {
            $this->Flash->success(__('The bill has been saved.'));
            return $this->redirect(['action' => 'index']);
        }
        $this->Flash->error(__('The bill could not be saved. Please, try again.'));
    }
    $companies = $this->Bills->Companies->find('list', ['limit' => 200]);
    $articles = $this->Bills->Articles->find('list', ['limit' => 200]);
    $this->set(compact('bill', 'companies', 'articles'));
    $this->set('_serialize', ['bill']);
}

Figure 12 Controller Layer for Bills

For example, here, depending on whether the user requests to view a certain bill or to add a certain bill, the view layer or the model layer will be called accordingly.

The architecture of MVC is as follows:
Implementation of The Solution

Back-Office Web Application

In this section, I will discuss the features and functionalities that the back-office web application offers, and I will do so by showing an example of a typical workflow operation.
1. By opening the website, the user is first prompted to enter his or her credentials:

2. If granted access, the user is redirected to this page from where he can access all the components of the system:
3. Supposing this is the first interaction with the system, the user can begin by adding clients to the database:
4. The process is the same for adding suppliers, articles, banks, and so on. Suppose we already have the following articles in the database:
5. The user can view, edit, or delete any article:

![Image of article management system]

6. Now, if the user wants to generate a bill to invoice a client, the following form will allow him to do so:

![Image of bill generation form]

The user chooses an article from the articles available in the stock, and the quantity needed. Automatically, the id or the code of that article is fetched, the price and the total
price as well. After validating the payment on a certain bill, the articles that we deployed or sold are deduced from the quantity on hand in the inventory. The same process applies for the purchasing component that involves suppliers instead of clients.

**Mobile Application**

The mobile application was developed in Android studio using a mixture of JAVA and PHP to communicate with the database. The application does not interact with the back-office web application, but deals directly with the database to preview and fetch data that the user wants to view.

Android Studio provides an interface that allows software engineers to develop mobile application in an organized way. Furthermore, this IDE provides an android emulator in case there is no compatible hardware available to test on. Drag and Drop features are provided and multiple layouts are available to allow maximum flexibility depending on the developer’s preferences.

While it is true that the IDE, if used with the emulator, is very heavy and requires a lot of processor power, resources, and performance, it assures a very stable development environment when the minimum requirements are provided. The snapshots in later sections show the aesthetics of the design as well as functionalities available.

**Platform architecture**

Android is an open source operating system for smartphones. It is Linux based and covers the widest range of devices in the market. This platform is a complicated one as it encapsulates many huge components grouped by category in the following diagram:
While not all of these components were used in the development of this mobile application, some of them were critical for the good functioning of the application. The system apps, which are the higher-level apps as shown at the top of the diagram, were all used for different purposes. The calendar API to keep track of the creation and expiration dates of Bills, Invoices, purchase orders, and so on.

**Testing**

The application was tested under different architectures and different version of Android, some of which are listed below:

- Sony Xperia M2 with Android 4.6
- Nexus 5X with Android 5
- Nexus One
- Pixel C

**Compatibility**

Because the application was developed using Java under Android Studio, it is most likely not going to work under all platform. Although the Android operating system covers the large portion of the mobile market, there remains others that the mobile application is not going to be compatible with such as iPhones or Windows Phones. To deal with that issue, a mirror of the mobile application will be developed using CakePHP 3 and hosted as a web application. This way, all devices will be able to access the platform and utilize its advantages without any significant or noticeable setbacks. The mobile application will be made responsive for maximum ease of use.

**Android Activity Workflow**

The following diagram shows the flow that an activity goes through when running the mobile application. An activity in android is described the view containing the personalized elements.
The activity abstract pre-implemented class provides multiple functions that allow the developer to implement actions when an event happens. As we can see from this diagram, multiple functions are available such as `onCreate()` which is when the frame is first open, or again `onPause()` when another activity is open on top of the existing one.

**Snapshots & Description**

The is the initial frame of the mobile application that the user has to use to authenticate. No options to recover the password are available using the mobile software, the admin has to be contacted in case loss of credentials.
These screenshots from the developed mobile application show an overview of the functionalities. From the first screenshot, we can see that the user is prompted for his credentials to log in to the system. Then, the main page the information of the company which can be edited from the mobile phone. The main page also shows a main menu that the user can use to navigate through the different parts of the application. The menu lets the user go to the Sales component or administrative component, or the contact component.

The contact component consists of 3 sub menus: Client, Supplier, and Employee. These three entities each have separate forms that allow viewing, sorting, or searching through the data. In some cases, as shown above, the user can even edit some information that is not critical such as modifying some personal information or some phone numbers. In the second part of the screenshots, the sale invoices are shown. They can also be viewed or sorted. The last screenshot shows an example of a bill where we can see specific information such as the total amount which is calculated from the existing articles lines.
The last screenshot also shows a horizontal view of the phone which demonstrates that the application is adaptable and responsive to any screen and any layout and responds dynamically depending on the hardware.

**Technological Enablers**

- **MySQL Server v5.7**
  This is the relational database management system used in the framework of this project.

- **MySQL Workbench 6.3 CE**
  This is the visual database design tool that was used to create and maintain the MySQL database.

- **CakePHP**
  The rapid development framework that was using to develop the back office using PHP with an MVC approach.
• **XAMPP**
  
  Was used to deploy and test the back-office website using the HTTP webserver apache.

• **Android Studio**
  
  The integrated development environment that was used to develop the mobile application.
Conclusion

The extend of this capstone project shows the importance of all the fields we study during our computer science bachelor here at AUI. Any entry level engineer would quickly realize that in the working world, it is not enough to be a good programmer or developer. To be an excellent and successful software engineer, or to lead in any other computer science field, there must be a certain level of communication with the clients and the shareholders. An engineer must be able to think about what suits the client best.

During the development of this project, I was able to tackle all aspects of computers science. From pure development to extensive communication, this project has prepared me for the professional world tasks. Project management, time management, rapid development techniques, new development paradigms: all skills that I had to enhance to complete this project successfully.