**Bits and Bytes - Planning Report**

Tiani Perera

Table of Contents

**Introduction2**

Introduction2

Aims of the Project2

**Initial Analysis3**

Interview Questions and Responses3

Refined Project Brief7

Function and Non-Functional Requirements and Constraints9

Use Case Diagrams11

**Identification of resources and Materials Required16**

UML Diagramming Tools16

Justification/Choice of UML Diagramming Tool16

Programming Language16

Justification/Choice of Programming Language17

Framework Options17

Justification/Choice of Framework18

IDE Options18

Justification/ Choice of IDE18

**Identification of Information Sources to be used19**

**Undertaking the analysis using the appropriate techniques20**

**Introduction**

Introduction:

I am building an upgraded IT based system for a company called Bits and Bytes that sells computer parts and computer systems. There are three main systems within the project; a stock control system for the products sold and any special one-time use voucher codes, a sales control system to manage sales, produce receipts and price lists, and a members control System to store details of the staff/member details and the special offers. Staff can use the sales control system to add products to the basket on behalf of customers, add any voucher codes and process payments.

Aims of the Project:

I will be following the Software Development Life Cycle – Agile Methodology to plan out my key Milestones. Using this methodology allows me to iterate any of the software development stages and ensure that each part of my system works. To do this, I will find any information necessary and use different resources to plan how I am going to build my application and what I am going to build it on. I would start by first analysing the project to better understand what is needed. For this stage to be successful, I will obtain a list of interview questions and answers, a detailed project brief and a list of requirements based on the brief. In addition to this, I will be building a use case diagram to better understand the customer’s needs.

The next step would be to design the system using diagrams such as Class Diagrams, Activity Diagrams and ERD Diagrams. These diagrams will help me in visualizing my system and its user interactions, which will then lead to a well-constructed application that will meet all the requirements of the brief.

After completing the planning and designing stages the next milestone would be the Development and Testing stage. This would mainly consist of coding my application. To help me achieve this stage, I will use the diagrams and documents I designed in the previous stages to ensure that I add all the key features necessary. After coding, I will methodically test my application and document my testing. This will help me in identifying any important missed criteria and update my system accordingly.

After the completion of all the stages, I will then move on to evaluating my software and evaluating how I met the criteria to hand in my application and produce an evaluation report. The evaluation report will contain all the details of this project along with the different steps I took to complete it. It will cover any changes made to the system between the planning stage and the evaluation stage, any hardships along the way and any improvements that could be made to the system.

**Initial Analysis**

Interview Questions and Responses:

1. I intend to use a logo to follow a colour scheme for the website. Any preferred colours or existing logos you would like me to consider for the website’s colour scheme?

We would like a Logo and colour scheme for each page / form in the application. However, we don’t have one as yet and would be comfortable with you designing something suitable using paint etc. You can also select the colour schemes.

Please use any colour consistently throughout all pages so a consistent feel is present.

We are far more interested in functionality however – so please make this feature a “nice to have” rather than essential priority.

1. Are there any features that are necessary that will help provide a user-friendly interaction?

We expect the application to avoid making the user unnecessarily click through options or menus when not necessary. We are looking for a slick, user friendly experience that is as efficiently designed as possible.

1. Which Systems can be accessed by which user roles?

There should be 2 staff roles – Sales Assistant, Manager.

**Sales Assistant** (access to view customer records, view all orders, makeorders for a customer, register a customers details (customers do not login – they are just listed in the same way stock is listed) , view stock, update stock)

**Manager** all functionality of Sales Assistant plus:

Adding new stock for sale

These next features are a “nice to have” rather than essential priority:

**Manager** should also be able to create new staff users and edit existing (i.e. change details and / or role).

**Manager** can access / run any reports if functionality is present for these

1. Other than the Store Manager and Assistant, are there any other user roles and can they access certain functionality (eg.Admin to manage page, moderator to manage user details)? Or is it just the two.

Only two for now

1. Are there any other systems in place (other than the stock control, sales and member control system), for example a system that allows staff to edit their own/members details, if so which user role can do access these systems

Staff should be allowed to edit a customer’s details on their behalf. Customers won’t have any direct access.

1. Is there a system need to allow the user in control of the stock system to order more products if stock is low, if so, what stock amount would be considered as low.

There is no need to “order” stock via this system. Our manual process of ordering stock from our suppliers will suffice for now.

When stock arrives we do need staff to have the ability to update stock counts for these products.

An additional feature that would be “nice to have” (but not essential) is that when stock hits 10 or fewer for any given product – an email is sent to the manager to remind them that we should create an order from the relevant supplier.

1. Are there any security measures for staff to log in.

If you have time – it would also be nice to see some kind of multi-factor authentication.

Something like an SMS or Email to double authenticate login is industry standard.

However, this is of low priority and not essential. So long as username and password works correctly, we are happy

1. What kind of product information and staff information do you need stored

We have provided a sample order form for assistance

In addition, we expect you to utilise common sense / common development approaches. We are open to solutions.

In terms of stock we expect a minimum of:

ProductID, product name, category, price, stock level and any other relevant (image if possible)

Customer: Email + standard user attributes

Staff: StaffID, password + standard user attributes + role, salary, dateHired

Order: OrderId, CustomerEmail, Order Date, Order Total + any other relevant

OrderLines should also be stored with standard attributes

1. What do you mean by customer Id and Membership Id? (e.g.: Do all members have both a customer Id and Membership Id, do all customers who are not a member have a customer Id when they buy a product)

These are mainly references to how our old (inefficient) system operated.

In the new application - Customers should automatically be enrolled as “members” when making a purchase if they haven’t already been registered. For security – all customers who shop at Big Vision must be registered. We do not allow anonymous purchases.

Customers will have an email address (unique) and a membership Id (also unique). Please treat references of Customer Id and Membership ID as one and the same

1. Are there any reports required (e.g. Total orders, total customers, sales for the day, stock reports), if so in what format and who can access them?

This is a nice to have feature and not essential.

However we would like managers to be able to run reports if possible. The formats we would accept are PDF, Excel, word, Any other valid. Your choice.

Reports we would like are:

All orders for a day

All stock that has less than 20 stock count

All orders for a customer

1. Is there any pre-existing date from your database.

We want a fresh start. With this in mind you will be asked to produce sample products / staff members etc to demonstrate the system functionality

1. When does the completed Bits & Bytes System need to be built by?

Design should be complete by 24th March and Coding and Testing should be complete by 26th May

The application needs to be fully completed (including testing and evaluation) ready for launch on Sunday 9th June at the latest (earlier is better)

1. What budget do you have in mind for me to hand you a complete functional system?

£59,000

1. Are there any specific payment methods you want included and excluded (example of payment methods could be cash, bank transfer, debit card, credit card, pay pal, google pay, stripe)

The **minimum** we require is a “simulated” payment system during the initial release of the software - where a form is displayed when a user selects to make their payment which provides the opportunity to input bank card, expiry, account name and ccv number in order to make the order as paid.

Ideally, if possible, we would prefer the system to link into one of the many payment gateway options out there such as Stripe, Paypal or any other relevant vendor

But if there are issues with this then please make sure the minimum is achieved

1. Are there any features that you would like to integrate in the future that will help scale your system?

At this time we are happy with what has been discussed. There are a number of “nice to have” features indicated above and if any / all of these are not present in the first version of the system that we receive we can discuss possible upgrade paths at that point

Refined Project Task:

Bits and Bytes current system requires the Manager to log stock in manually for assistants and a backup copy and requires the staff to manually write out customer receipts. I am required to build a new upgraded system with better functionality.

To use the new system, all staff must login, multi-factor authentication will be used as well as a username and password for security purposes. Once they have logged in, they will be able to edit their own details and change their password.

The new stock system must store information on the products, it should contain a unique description of each product along with the price and stock level. Stock is divided into three categories: Computer Parts, Computer systems and special offers. The stock system will have a feature that alerts the manager via email if stock level is 10 or lower for any product, this will allow them to manually create a stock order. Staff must be allowed to view and update stock level for computer parts and computer systems, they should be able to do this via a form on the system. Managers can add new stock, amend, delete them. Stock will automatically be updated when an order goes through. Special offers are voucher codes (5% off, 10% off, 15% off) that will be stored in the database, they can be created by managers, and be viewed by all staff. Voucher codes can be emailed out to customers and can only be used once.

The sales control system allows staff to attend to customer orders. Once the customer has given their details, staff can add products that are available for purchase. Once the staff has completed adding the customers’ products and any special voucher codes, they will be taking to a payment page where a simulated payment system is set up. When the user selects the bank card payment method, they will be shown a form to input their bank card, expiry date and account name along with the cvv number. The system should be most likely to link into one of the many payment gateway options such as Stripe, PayPal or any other relevant vendor. When the payment has been authenticated and the transaction is complete, an email confirmation will be sent out to the customer. The receipt will have customer information, product information, Vat, the date, and the name of the staff who served the customer. Staff can view/save/print these receipts. In case a payment doesn’t go through, staff can always choose a different payment method for the customer. Staff can also view all orders placed via the sales system.

The member’s control system allows staff to keep track of member details and discount status. Every customer that makes a purchase from the company must give their details so that the staff can register them in the system as a member. No anonymous purchases are allowed. Each customer will be asked to give their email address, name, address, phone number, Date of birth (not compulsory), they are then presented with a unique membership ID. Email addresses provided used must be unique within the system. Staff can also view and update member details on their behalf, to do this, customers will receive a unique code via email or SMS for security purposes. The discount status represents the customer’s status which depends on a bonus scheme that allows members to receive a permanent 5% discount voucher when they have shopped 30 times, 10% discount voucher when they have shopped 60 times, or a VAT Free voucher (20% discount) when they have shopped 90 times. This should automatically be added on the view basket page.

Managers are allowed to create new staff and they can view all staff details including their role, salary and date hired. Managers can also change staff roles. Managers can also view reports for all orders placed on that day, all stock that has less than 20 pieces, and all orders for a certain customers. These reports can be viewed as a pdf, excel or word document.

Each order must include the order Id generated by the program, the current date, the staff name and ID, it should also contain customer information along with the product information and the order total. Each stock must have a product ID, name, category, price, stock level and image. Customers should have a unique email address, unique membership ID and all other customer information such as phone number etc. Staff must contain its ID, password, their standard user attributes along with they role, salary and the date they were hired. The database must be seeded with sample products, staff members and customers to showcase the functionality of the system. The order total must be calculated by adding the total of (no of items\* price of product) plus 20% VAT.

Functional and Non-Functional Requirements and Constraints:

Functional Requirements:

System

* Should have links and menus to navigate through the system
* Must have a staff log In page
* Must allow staff to login with their username and password along with multi-factor authentication
* The system must allow different staff roles : Manager and Sales Assistant
* Each role has their unique navigation links based on the requirements
* The system must have a stock system, a members control system along with sales system
* The stock system must have products of three categories: Computer Parts, Computer Systems and special offers
* When the stock levels for a product is below 10, the manager is alerted via email
* Stock level automatically updates itself when an order is processed
* Must have a stimulated payment feature that produces receipts
* 20% VAT must be added to all orders
* The system must accept special voucher codes, that can only be used once
* The system must have a feature that checks how many purchases a customer has made and sets their account with a permanent discount status (5% off for 30+ shops, 10% off for 60+ shops and VAT Free for 90+ shops)
* Must have a view basket page
* Must have a view products page

All Staff(Sales Assistants and Managers)can

* View customer orders
* Create customer orders
* Register customer details
* Add customer information to the order
* View customer details
* Edit customer details with 2 factor authentication
* View voucher codes
* Add voucher codes to a customer order
* View Stock Level
* Update Stock Level
* Edit their own details
* Change their password
* Process a payment for a customer with the customers bank card, expiry date, account name and cvv
* Print receipts / View Receipt
* Customer Receipts must include

Only Managers can

* View/Print/Download Reports: Daily Transactions Report, Low Stock Report and Customer Report
* Create New Staff and edit their details
* Delete Staff Accounts
* Change Staff Role
* Add new stock, amend and delete them
* Create new voucher codes – 15% off, 10% off and 5% off
* Send unique voucher codes to customers via email

Non-Functional Requirements:

* The application should have a consistent colour scheme and logo
* The System must meet security standards (Two-Factor Authentication for user login, lock account after 3 unsuccessful tries)
* The system must allow users to use the forgot password feature
* The system must have a back-up copy in the event of corruption of the original system
* The system must avoid any unnecessary click through options/menus
* The system must be lightweight, user friendly and efficiently designed

Constraints:

* Must not go over the budget of £59,000
* All designing of the application must be completed by the 24th of March 2024
* Final fully functional and tested application must be completed by the 9th of June 2024

Use Case Diagram along with Examples:

(see end of document for better view)

~~A diagram of a diagram

Description automatically generated~~

|  |
| --- |
| **Use Case Documentation** |
| **Use case name: Login** |
| **Description: Logs the user in to their account** |
| **Initiating Actor / Trigger Event: Staff** |
| **Pre-conditions: Staff has launched the Bits and Bytes System** |
| **Normal scenarios (Success Path):**   1. **User enters username** 2. **User enters password** 3. **User clicks LOGIN button** 4. **User gets code via email(2 factor authentication)** 5. **User enters the correct code** 6. **User is logged in** |
| **Abnormal scenarios (Extensions):**  If username and / or password are incorrect and customer cannot remember their details then:   * **Execute Reset Password Use Case**   If username and / or password has been entered incorrectly 3 times in a row:   * **Execute Lock Account Use Case** |
| **Includes scenarios:**  **None applicable** |

|  |
| --- |
| **Use Case Documentation** |
| **Use case name: Make Payment** |
| **Description: Customer pays for Order** |
| **Initiating Actor / Trigger Event: Staff** |
| **Pre-conditions: Staff has clicked on Pay for order button** |
| **Normal scenarios (Success Path):**   1. **Staff sees the final price (after any discounts)** 2. **Staff selects the choses customer payment type** 3. **Staff enters any bank details and presses the pay now button** 4. **Authentication of payment is made through third party – bank/PayPal/stripe** 5. **Payment is completed** |
| **Abnormal scenarios (Extensions):**  **If payment doesn’t go through due to an authentication failure**   * **execute Make a Payment, to allow staff choose a different payment method for the customer** |
| **Includes scenarios:**  **None Applicable** |

|  |
| --- |
| **Use Case Documentation** |
| **Use case name: Add Products** |
| **Description: Allows the user to add products the basket** |
| **Initiating Actor / Trigger Event: Staff** |
| **Pre-conditions: Customer information must be entered** |
| **Normal scenarios (Success Path):**   1. **User selects a product** 2. **User chooses a quantity** 3. **User clicks add to basket button** 4. **Product is added to the basket** |
| **Abnormal scenarios (Extensions):**  **None Applicable** |
| **Includes scenarios:**  **None Applicable** |

|  |
| --- |
| **Use Case Documentation** |
| **Use case name: Get customer Information** |
| **Description: Get customer Information for the order** |
| **Initiating Actor / Trigger Event: Staff** |
| **Pre-conditions: User must be logged in** |
| **Normal scenarios (Success Path):**   1. **User enters Customer email** 2. **Account details are found** 3. **Products can be added to basket** |
| **Abnormal scenarios (Extensions):**  If Customer doesn’t have an account created then:   * **Execute Create customer Account Use Case** |
| **Includes scenarios:**  **None applicable** |

**Identification and Procurement of Resources and Materials Required**

UML Diagramming Tools:

Visual Paradigm is an application that offers tools for software development and system design. Visual Paradigm comes as a desktop application that needs to be downloaded and set up, or as an online tool. They offer a free commercial edition along with other paid more advances versions of paradigm. They offer a range of diagram types such as Class Diagrams, Activity Diagrams and Entity-Relationship Diagrams. You select the type of diagram you want to build, and it displays a toolbox with the relevant tools for that diagram. Visual Paradigm is extremely flexible and has collaboration features, which allows many members to work on a project. However, for a beginner user it may be complicated to use their tools.

Lucid Chart is an online tool that aids the creation of diagrams and charts, they also have a mobile application for smartphones and tablets. They offer a free version and a paid subscription to unlock all the features of this tool. They too offer a range of diagram templates including wireframes and flowcharts. Lucid Chart is known for being extremely user-friendly and it very simple to use. However, on the downside, the free licence for Lucid Chart comes with a very limited number of features.

Justification of Choice of UML Diagramming Tool:

For my project I will be using Visual Paradigm to draw up my diagrams. It seems to offer a more specialized experience based on the diagram type I have chosen. After trying out the app, I prefer the online tool better and after many class hours of using Visual Paradigm, I am confident in my abilities to navigate through the webpage and successfully complete my diagrams.

Programming Language:

C# is a modern object-oriented programming language that can be used to build many types of applications, and can also support different concepts including classes, objects, polymorphism, etc. C# programs were originally built to run on .NET framework but has further been modified to run on multiple platforms such as Mono and Xamarin. C# offers and extensive standard library that consists of classes/functions for database access, etc. The NuGet package manager can also provide third-party libraries and frameworks.

This programming language is commonly used to build interactive web applications. Frameworks including ASP.NET and ASP. Core can be utilized to build them. In addition to this, C# can also be used in game development, mobile-app development, and cloud-based applications. The syntax of C# is similar to languages such as C++, and consists of expressions, statements, blocks of code, etc. There are special keywords used to define classes (e.g. class), datatypes (e.g. var), etc that cannot be used as identifiers.

C# is seamlessly integrated with Windows which allows users to make use of technologies such as Windows Forms. It contains features such as LINQ that allow code to be clean and maintainable. It is supported by IDEs such as Visual Studio Code; this offers additional tools such as debugging to be used. However, if building a cross-platform application, it may need additional effort to allow the transition.

Java is a secure and reliable object-oriented language that can be used for coding various types of applications from mobile applications to server-side technologies. Java is a language that has been used consistently and has a rich ecosystem of tools. It contains many in-built functions, libraries, etc, that would help build this range of applications. Java is simple and easy to understand, it can be broken down into methods, classes, and features, It can run only any platform given that it has a JVM (Java Virtual Machine). IDEs such as NetBeans are used for development and they provide additional resources such as debugging and refactoring.

Java allows applications to be run on diverse environments with no additional effort and consists of many functional tools. However, Java may require more lines of code, which indicates a longer development period. Compard to modern programming languages, it may have a lower performance level.

Justification of Choice of Programming Language:

As I am familiar with both programming languages, I have decided to proceed with C#. C# consists of more modern language features and seems to have better performance. I am already well-revised on its syntax and features, and can use my existing knowledge to better develop a high-quality conclusion.

Framework Options:

MVC(Model-View-Controller) is used in web application development. The model contains the logic of the application. It communicates with the database to perform CRUD operations. The view presents the user interface by retrieving data from the model. The controller receives the user input from the view and processes the data with the help of the model and updates the view. MVC allows each component to be individually tested, which assists the programmer in building a more reliable application. The models, views and controllers can be used across multiple segments of the application. This will consequentially reduce the development time. On the other hand, MVC does have a steep learning curve, causing most developers to have a slower start to developing.

Windows Forms is a GUI (Graphical user Interface) that consists of pre-built controls and components that can be used when building the user interface. Windows Forms allows for rapid development with their drag-and-drop interface. It is also seamlessly integrated with .NET Framework. However, applications that use Windows Forms do have limited cross-platform support and can also be difficult to maintain and scale due to its lack of architecture.

Justification of Choice of Framework:

I will be utilizing MVC as it is more flexible, easy to test and is more suited for web-development. This will aid me in making my coding journey simple. MVC seems to have a large and active community of developers along with multiple tutorials, which can facilitate in troubleshooting my application and learning new material. Having coded using the MVC framework gives me a better understanding of how to work with the MVC Architecture.

IDE Options:

Visual Studio Code is a lightweight and versatile Microsoft code editor. VSC is a free, open-source application available for download. It offers support for debugging, task running, version control, etc. It also has good cross platform compatibility which promotes collaboration and consistency across various OS. Visual Studio Code can support many programming languages including C#. Its lightweight nature allows for a quick and easy start-up. While VSC has many advantages, it does lack certain features and tools. It does not offer the same level of support for .NET development compared to Visual Studio.

Visual Studio is an extensive IDE and code editor specifically made for software development with Microsoft technologies such as ASP.NET and C#. Visual Studio provides a wide spectrum of tools for development. Its advanced features allow developers to have everything they need in one development environment. Visual Studio is also best for large scale projects. On the downside, its advanced set of tools results in a steep learning curve. Visual Studio may need powerful hardware which leads to greater start up times and higher memory being used. It is available to use as a free community edition or as a paid professional version which provides additional features.

Justification of Choice of IDE:

I have chosen to work with Visual studio Code due it its lightweight nature and high performance. VSC also has a built-in Git Integration, which will make it easy to work with Git. Additionally, having worked with both Visual Studio and Visual Studio Code, I seem to prefer VSC over VS.

**Identification of Information Sources to be used**

**Fundamentals of Computer Programming with CSharp (2013)** by Nakov will assist and guide me in learning more about my programming language. It covers the basic knowledge needed to program in C#, in addition to this it also contains a wealth of practical examples, questions and advice. This will be an invaluable resource in building my application.

**GitHub** is web-based platform that allows developers to store, track and collaborate on different projects. It has a large online community and is easy to use for beginners. It consists of many learning resources and will allow me to access specific resources related to any errors I might have.

Source: <https://github.com/?ef_id=_k_CjwKCAiAlcyuBhBnEiwAOGZ2SxnKp3IJ4fXJtZ1bzVDKW_YASOVIQQNO4GHQn9Z7dW2OjlNwCABymhoCjewQAvD_BwE_k_&OCID=AIDcmm153qff2f_SEM__k_CjwKCAiAlcyuBhBnEiwAOGZ2SxnKp3IJ4fXJtZ1bzVDKW_YASOVIQQNO4GHQn9Z7dW2OjlNwCABymhoCjewQAvD_BwE_k_&gad_source=1&gclid=CjwKCAiAlcyuBhBnEiwAOGZ2SxnKp3IJ4fXJtZ1bzVDKW_YASOVIQQNO4GHQn9Z7dW2OjlNwCABymhoCjewQAvD_BwE>

**Stack Overflow** is a question-answer style website that allows developers to ask and answer programming questions. I will mostly be using this site when I encounter any errors during my development process. It already has many common questions and solutions on this page, which may be useful. This is also an invaluable resource as it allows me to browse through other code and deepen my understanding.

Source: <https://stackoverflow.com/>

**Microsoft Documentation for ASP.NET MVC** is the official Microsoft website that provides multiple guides, videos, and examples. It also includes API references for different classes, methods, attributes, etc. This will provide me with the guidance needed to produce a well-rounded application.

Source: <https://learn.microsoft.com/en-us/aspnet/mvc/>

**Programming with Mosh** is a YouTube channel that contains tutorials on different aspects and features of coding, including ASP.NET MVC and C#. There are also many other YouTube channels that will assist me in coding, but also adding any new features and making my application more user friendly.

Source: <https://www.youtube.com/c/programmingwithmosh>

**Undertaking the Analysis using Appropriate Techniques**

I utilized the brief given to me and combined it with my previous experiences of building applications to clarify the content of the brief and identify any other content that may be useful. I then complied a list of questions to hand to the owner of the company. The answers to these questions helped me clarify any doubts I had and assisted me in addressing the owner’s needs.

I drew up the use cases on visual paradigm based on the requirements of the system. Designing the use case diagram assisted me in better understanding the user’s journey in making a customer order. It made me look at possible features that need to be added and identify anything that needed to be addressed.

I conducted my research on google and YouTube to assist me in selecting the most suitable choice of development platform. I browed through various documentation for VS and C# to understand its capabilities. I read through books and used YouTube to help me see how I could possibly implement certain features on various platforms. Most of the code to these features could also be found of stack overflow.

The Agile Methodology allows me to create small sections of my application, test its functionality and then move on to the next. This is extremely flexible and allows me to continuously improve my project. The Agile Coach (Atlassian’s coaching website) offered an in depth description of everything I need to know about this methodology to help code my application.

~~A diagram of a diagram

Description automatically generated~~