

Project Proposal

Sales Reporting and Prediction System

<u>MEMBERS</u>	<u>STUDENT ID</u>
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]
Marcus Rakkhit	[REDACTED]
[REDACTED]	[REDACTED]

Tutorial - Monday 2:30pm EN402 with [REDACTED]

Background / Problem Description

We have been hired by People Health Pharmacy (PHP) Inc. to create software that will allow them to better understand their sales over a weekly and/or monthly basis. In doing so, they will ultimately be able to make informed decisions about their stock ordering procedure.

Scope

The software shall provide a graphical interface to generate sales data from a database; This should allow the company to understand the weekly and monthly sales. Users will also have the ability to add, display, update and remove items from the database as needed with certain privileged users able to edit the database further (this will allow employees to analyse the data); This will be enforced using an authentication system to ensure that security is implemented in the application.

In regards to the quality of scope, the brief specified to have a graphical user interface; In response, our group can download a library package (depending on the programming language we choose) that allows us to create a graphical interface.


The software will also have the functionality to export this data into a CSV file for further processing, however this further processing is out of the scope of this project and the software. Furthermore, the clients can choose what records can be on the database (e.g. they can choose their own products to add to the database) and make their own decisions to refill products.

Though we will be creating the database, we will not be responsible for adding the data, determining the type of data, or maintaining the database after the project's completion.

The budget of the project should be \$0 and each sprint period should last around 2 weeks.

Stakeholders

People Health Pharmacy (PHP) - Company paying for the new software to be used at their store.

Team Sponsor  - Sponsoring the team to make the program.

Project Team - Making the program to work within the Pharmacy.

Deliverables and schedule

The only deliverable for this project will be the program that we are completely for People Health Pharmacy. The details of this program and its product backlog items are listed below.

Initial Release Schedule

Product Backlog Id.	Product Backlog Item	Dependencies (Product Backlog Id.)	Business Value (1 least – 10 most)	Release Schedule (Sprint #1 2 3 ...)
1	Usable Graphical Interface		7	Sprint #1
2	Add a record to the database		6	Sprint #1
3	Edit a record in the database	2	6	Sprint #1
4	Display a record from the database	1, 2	7	Sprint #1
5	Show the weekly/monthly sales report	1	9	Sprint #1
6	Predict the weekly/monthly sales of an item	2	9	Sprint #1
7	Predict the weekly/monthly sales of a group of similar items	2	9	Sprint #2

8	Generate a weekly/monthly sales report in a CSV file	1, 2	9	Sprint #1
9	Notify on low quantity of stock	1	9	Sprint #2
10	Create user types		2	Sprint #2
11	Create privileges for each user type	10	4	Sprint #2
12	Validation of users (e.g. password)	10	3	Sprint #2





Solution Direction

In this project, 2 solutions were proposed: A desktop application and a web-based application.

The problem domain consists of business operations, economics and the general products the business sells at the Pharmacy store.

The solution domain consists of sales records, data reports, notifications, mathematical calculations, and database CRUD operations

SKILLS

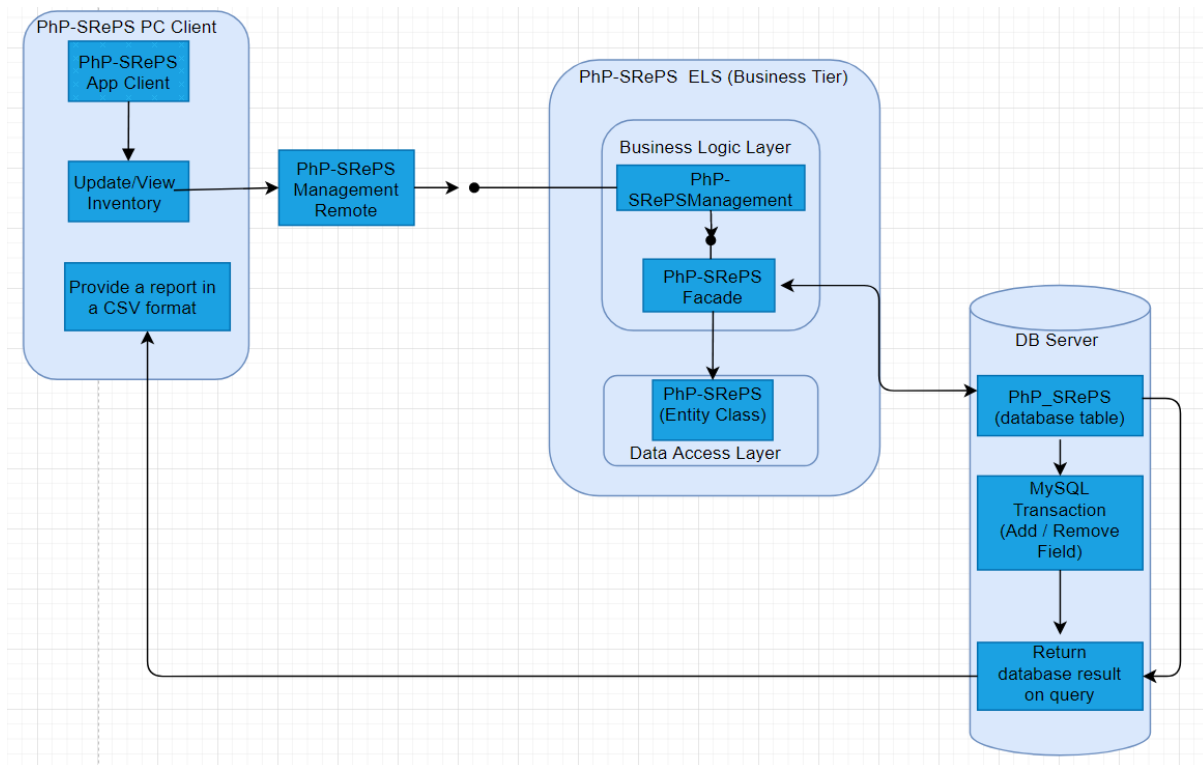
Members	Skills
	<ul style="list-style-type: none"> • Experience in object oriented languages such as C#, Java and Kotlin • Knowledge with web technologies such as HTML, CSS, PHP, mySQL etc. • Familiar with UNIX based systems such as Linux and their shells such as bash and zsh
	<ul style="list-style-type: none"> • Prior knowledge with topics covered in developing web applications (HTML,CSS, PHP and JavaScript) • Worked with databases and queries • C, C++
Marcus	<ul style="list-style-type: none"> • For desktop application, I've done MySQL databases • Did Object Oriented Programming • I've covered HTML, JAVASCRIPT, PHP in Creating Web Applications • Diagrams down below covered similarly to the Java EE diagrams in Creating Secure and Scalable software
	<ul style="list-style-type: none"> • Previous experience with SQL databases, website programming [HTML, CSS, Javascript], C# • Prior experience in retail stores
	<ul style="list-style-type: none"> • Familiar with Object Oriented Programming • Some knowledge in Web development using HTML, CSS, JS, and PHP • C# • SQL Databases and queries • AWS Cloud technologies

COMPARISON

Solution	Rationale	Advantage of Solution	Disadvantage of Solution
Desktop Application	Desktop applications are optimal for deploying to small businesses. Desktop applications allow high efficiency for deployment of its software. The database is stored on the local network; thus providing more security for data access.	<ul style="list-style-type: none"> - Very optimal for operating in small business environments - Very easy to deploy to existing business infrastructure 	<ul style="list-style-type: none"> - Applications can only operate on a PC - Not capable of running on many operating systems
Web Interface	The web interface allows the software to be deployed on multiple platforms; The main issue for this solution is the potential for cyber-attacks.	<ul style="list-style-type: none"> - The application can run on different devices with varying operating systems - Only needs to be written once for all platforms - Portable: Remote devices such as laptops/tablets/phones can utilize it 	<ul style="list-style-type: none"> - System may not work if the internet is unavailable - Risks of cyber-attacks (cross site scripting- attacker can masquerade as user, burp to bypass client side validation etc.)

We believe going with a desktop application is the best route for this project. Having a web app requires more knowledge of web technologies and a multitude of languages which may make the project more complex. It also means that the software won't rely on internet access to process requests and generate reports.

SOFTWARE DIAGRAM



PHP-SRePS PC Client:

- **PHP-SRePS App Client:** Allows clients to test the CRUD operations
- **Update/View Inventory:** Contains form needed to start the updating/viewing process
- **Provide a report in a CSV format:** Creates a CSV file based on the returned result

PHP-SRePS Management Remote: Waits for a requests from the business tier

PHP-SRePS ELS (Business Tier):

- **Business Logic Layer:**
 - **PHP-SRePS Management:** Determines if logic is sufficient
 - **PHP-SRePS Facade:** Methods that executes CRUD operations to and from the database
- **Data Access Layer:** Contains all attributes from database concealed inside one object class

DB Server:

- **PHP-SRePS (database table):** A table that Contains all records
- **MySQL Transaction (Add/Remove Field):** adds records or removes records from the table





Return database result on query: Returns results to the PC Client

Quality Management

<u>Backlog Item</u>	<u>Definition of Done</u>
<u>Setup the Database</u>	A database should be set up that can create multiple objectives with different attributes inside of it. This database should be able to contain all the necessary requirements that the company needs with their sales, including pricing, items, stock, etc.
<u>Setup/Program the Graphical Interface</u>	Graphical interface should be completely accessible for users to use.
<u>Add a record to the database</u>	Records should be added to the database correctly, with each field filled out as is required.
<u>Edit a record to the database</u>	Records inside the database should be able to be edited to change parts of the record to the correct amounts.
<u>Display a record from the database</u>	A record should be viewable on the Graphical interface.
<u>Show the weekly/monthly sales report</u>	The graphical interface should be able to show the weekly / monthly sales of a specific item based on a selected time frame.
<u>Predict the weekly/monthly sales of a group of similar items</u>	Using the weekly / monthly sales of an item, this software should be able to predict the sales of said item by calculating how popular that item is
<u>Generate a weekly/monthly sales report in a CSV file</u>	The sales report that the program can create should be able to be printed out into a CSV file that contains all the information inside it.
<u>Notify/refill products on demand</u>	When a specific item would be assumed to run out in the next 2 weeks based upon the weekly/monthly sales report, the program should be able to notify the user of this refill as well as go through the process of ordering the

	extra stock.
<u>Create user Types</u>	Different user types should be able to be created.
<u>Create privileges for each user type</u>	Limit access to particular methods for specific user types (eg. only administrator user types can add new users), otherwise present a dialog notifying the user that they do not have permission to perform the current task
<u>Validate users (e.g passwords)</u>	This will be completed when the program allows the user to enter a user account with a password input that matches that account's password.

Resources

<u>Team Member</u>	<u>Role</u>
	Editor, Developer
	Developer
	Developer
	Back-End Developer
Marcus Rakkhit	Database Developer, Facilitator/Team Leader

Approval Signatures:

Project Team

	Name of Student	Student ID	Signature
1	[REDACTED]	[REDACTED]	[REDACTED]
2	[REDACTED]	[REDACTED]	[REDACTED]
3	[REDACTED]	[REDACTED]	[REDACTED]
4	[REDACTED]	[REDACTED]	[REDACTED]
5	Marcus Rakkhit	[REDACTED]	[REDACTED]

Project Sponsor [Your Tutor]

Tutor's name (on behalf of the client)	Signature:
[REDACTED]	
