SWINBURNE UNIVERSITY OF TECHNOLOGY

HO CHI MINH CAMPUS

COS20031 - Computing Technology Design Project

Topic

PROGRESS REPORT

FOR B2B2C ECOMMERCE PLATFORM SYSTEM

INSTRUCTOR: Hang Sam Nang

STUDENT NAMES:

Nguyen Quoc Thang

Le Huu Tuong

Tran Thanh Minh

Trinh Quy Khang

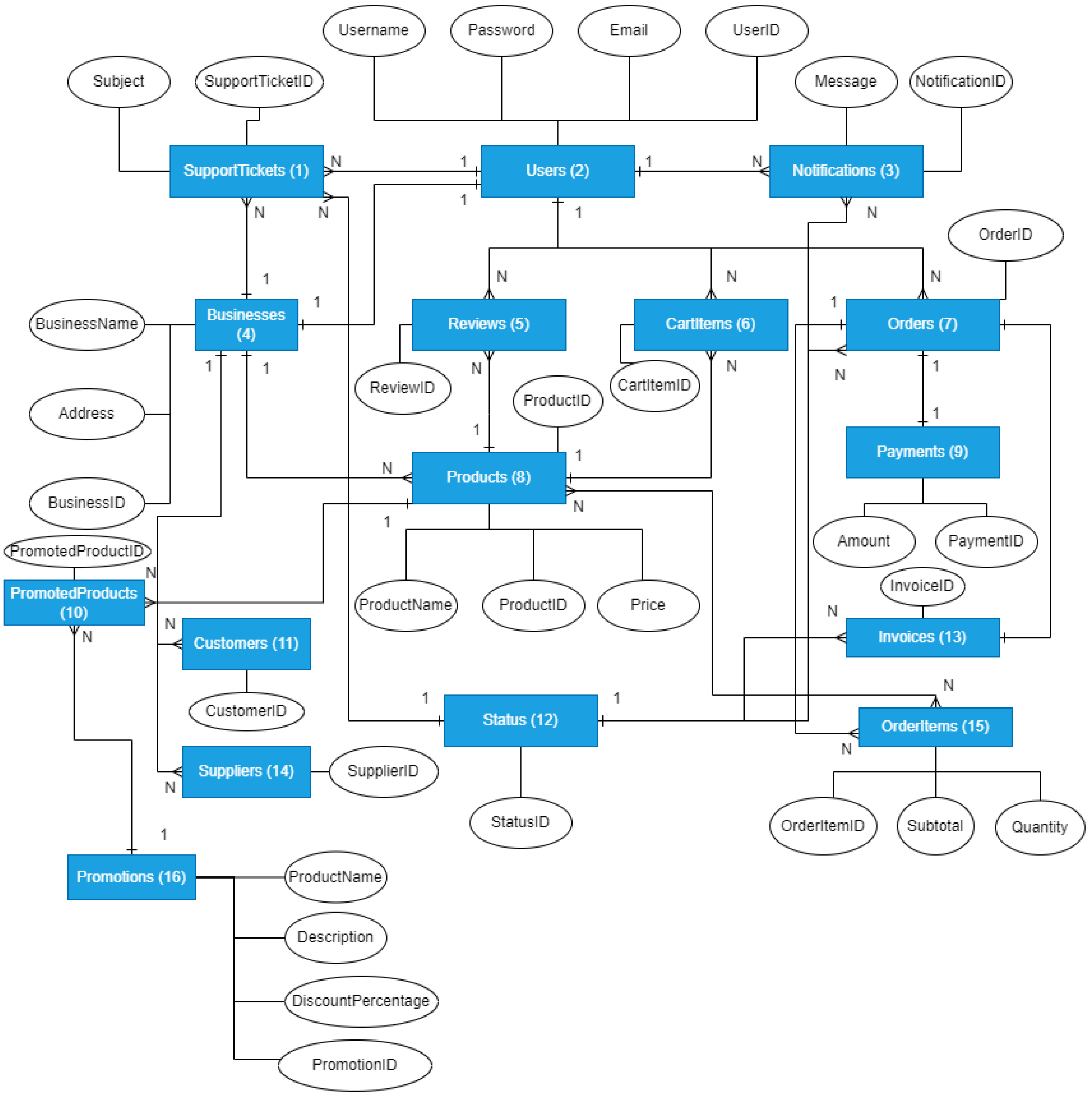
Nguyen Do Nhat Nam

Tran Thai Duy Khang

HO CHI MINH CITY – October 2023

Initial ER Diagram

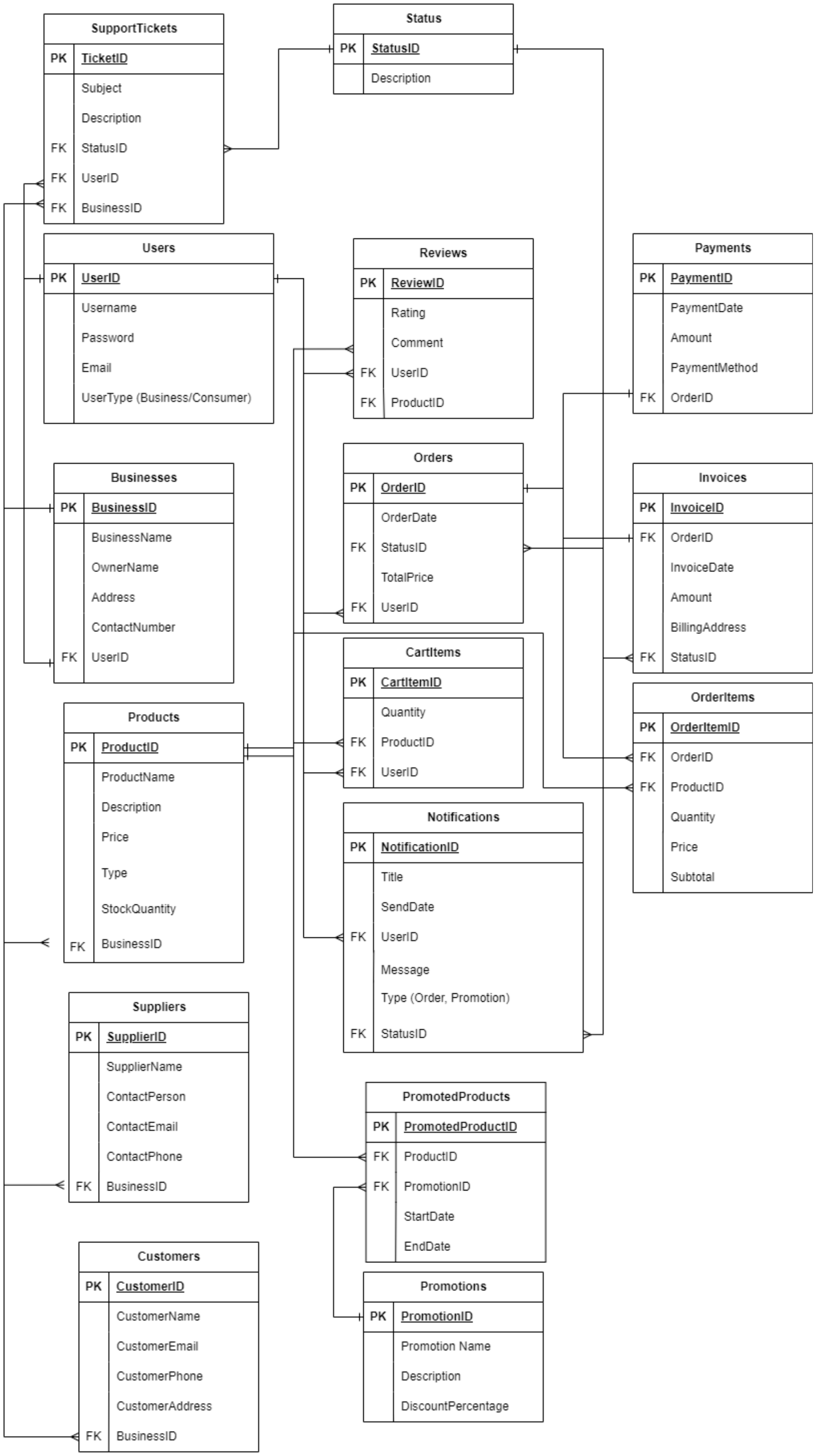
Concept Relationship Diagram



Concept Relationship Diagram

Entity Relationship Diagram

Note: Foreign Key (FK), Primary Key (PK)



Explanation of ERD

Here are the explanation of the use of different entities in the ERD and later on will be used for designing the database for the Ecommerce platform system:

- **Users**: The User table will contain all the credential information of the consumers and businesses for logging into the B2B2C Ecommerce platform system.
- **Businesses**: This table will house the essential business details, allowing the platform owner to validate whether a business is suitable for using the platform. The table is also uniquely identified by UserID as foreign key.
- **Products**: Within this table, comprehensive product details will be stored. Both businesses and consumers can access this information to make purchase decisions. The Product table is further identified by the BusinessID as a foreign key, providing clarity on the product's owner.
- **Orders**: The Order table serves as a repository for order-related details from all clients. Orders can be organized based on their status, and each order is associated with a UserID, providing insight into the ordering user.
- **CartItems**: This table will maintain records of all items stored in a user's shopping cart. It is identified by both the ProductID and UserID as foreign keys, allowing for the tracking of which user placed the order and which product is involved.
- **Reviews**: Within this table, user reviews, ratings, and comments about various products are stored. Reviews are linked to users and products through the UserID and ProductID foreign keys, facilitating a comprehensive feedback system.
- **Notifications**: The Notification table will house all user notifications and is associated with users via the UserID foreign key.
- **Payments**: This table will store payment details for each order, with the OrderID serving as the foreign key, connecting payments to specific orders.
- **SupportTickets**: This table is designed to record all user-generated support requests. The optional BusinessID foreign key allows for associations with businesses when applicable.
- **Customers**: This table is designed to record the customers who have bought the

products of the business and use the BusinessID for the foreign key.

- **Suppliers**: This table is designed to record the multiple suppliers of the business.
- **OrderItems**: This table is designed to record the ordered item's details.
- **Invoices**: This table is designed to store the detailed invoice of each order.
- **Status**: This table is designed to store all the status descriptions of other tables.
- **PromotedProducts**: This table is designed to remove the conjunction between table Products and table Promotions and can keep track of the promotion

Explanation of relationship in the ERD:

- **Users - Businesses (One-To-One)**: Only one user can be associated with one business
- **Businesses - Products (One-To-Many)**: One business can have multiple products.
- **Users - Orders (One-To-Many)**: One user can place multiple orders.
- **Users - CartItems (One-To-Many)**: One user can have multiple cart items.
- **Users - Reviews (One-To-Many)**: One user can write multiple reviews and ratings.
- **Users - Notifications (One-To-Many)**: One user can receive multiple notifications.
- **Orders - Payments (One-To-One)**: One order can only have one payment.
- **Users - SupportTickets (One-To-Many)**: One user can create multiple support tickets.
- **Businesses - SupportTickets (One-To-Many)**: One business can be associated with multiple support tickets and is requested from the consumers so this column is optional.
- **Businesses - Customers (One-To-Many)**: This relationship signifies that a single business can have numerous customers, reflecting the fact that businesses cater to a wide clientele.
- **Businesses - Suppliers (One-To-Many**): In this context, the relationship denotes that one business can establish connections with multiple suppliers. This illustrates that businesses often engage with multiple entities to source their products or materials.
- **Promotions - PromotedProducts (One-To-Many)**: This association demonstrates that one promotion can apply to multiple products. This showcases the flexibility of promotions being applicable to a range of products.
- **Products - PromotedProducts (One-To-Many)**: This association demonstrates that one product can be associated with multiple promotions.
- **Orders - OrderItems (One-To-Many)**: This relationship signifies that a single order can encompass multiple items. It underscores the idea that an order can consist of various products or services.
- **Orders - Invoice (One-To-One)**: In this scenario, each order is uniquely associated with one invoice. This ensures a one-to-one correspondence between an order and its

corresponding invoice, simplifying the billing process.

- **Products - OrderItems (One-To-Many)**: This association illustrates the versatility where one product can be included in multiple order items. This highlights the dynamic nature of product inclusion in various order items.
- **Status - SupportTickets (One-To-Many)**: This association links the description of the status of the SupportTicket table.
- **Status - Invoices (One-To-Many)**: This association to link the description of the status of the Invoice table.
- **Status - Orders (One-To-Many)**: This association to link the description of the status of the Order table.
- **Status - Notifications (One-To-Many)**: This association to link the description of the status of the Notification table.

SQL Query

Here are all the SQL queries for creating tables in MySQL, the order has been organized so that it need to be configured in this order otherwise the errors will occur:

1. **Table Status**:

   ```
   CREATE TABLE Status (
     StatusID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
     Description CHAR NOT NULL
   );
   ```

2. **Table Users**:

   ```
   CREATE TABLE Users (
     UserID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
     Username VARCHAR(30) NOT NULL,
     Password VARCHAR(20) NOT NULL,
     Email VARCHAR(30) NOT NULL,
     UserType VARCHAR(20) NOT NULL
   );
   ```

3. **Table Promotions**:

   ```
   CREATE TABLE Promotions (
     PromotionID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
     PromotionName VARCHAR(55) NOT NULL,
     Description VARCHAR(255) NOT NULL,
     DiscountPercentage DECIMAL(10,2) NOT NULL
   );
   ```

4. **Table SupportTickets**:

   ```
   CREATE TABLE SupportTickets (
     TicketID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
     Description LONGTEXT NOT NULL,
     Subject VARCHAR(255) NOT NULL,
     StatusID INT,
     FOREIGN KEY (StatusID) REFERENCES Status(StatusID) ON DELETE CASCADE,
     UserID INT,
     FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE
   ```

```
);
```

5. **Table Businesses**:

```
CREATE TABLE Businesses (
  BusinessID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  BusinessName VARCHAR(30) NOT NULL,
  OwnerName VARCHAR(30) NOT NULL,
  Address VARCHAR(30) NOT NULL,
  ContactNumber VARCHAR(10) NOT NULL,
  UserID INT,
  FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE
);
```

6. **Table Customers**:

```
CREATE TABLE Customers (
  CustomerID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CustomerName VARCHAR(30) NOT NULL,
  CustomerEmail VARCHAR(30) NOT NULL,
  CustomerPhone VARCHAR(10) NOT NULL,
  CustomerAddress VARCHAR(30) NOT NULL,
  BusinessID INT,
  FOREIGN KEY (BusinessID) REFERENCES Businesses(BusinessID) ON DELETE CASCADE
);
```

7. **Table Products**:

```
CREATE TABLE Products (
  ProductID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  ProductName VARCHAR(30) NOT NULL,
  Description VARCHAR(255) NOT NULL,
  Price DECIMAL(10,2) NOT NULL,
  Type VARCHAR(20) NOT NULL,
  StockQuantity INT(255) NOT NULL,
  BusinessID INT,
  FOREIGN KEY (BusinessID) REFERENCES Businesses(BusinessID) ON DELETE CASCADE
 );
```

8. **Table Suppliers**:

```
CREATE TABLE Suppliers (
  SupplierID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  SupplierName VARCHAR(30) NOT NULL,
  ContactPerson VARCHAR(30) NOT NULL,
  ContactEmail VARCHAR(30) NOT NULL,
  ContactPhone VARCHAR(10) NOT NULL,
  BusinessID INT,
  FOREIGN KEY (BusinessID) REFERENCES Businesses(BusinessID) ON DELETE
CASCADE
);
```

9. **Table Reviews**:

```
CREATE TABLE Reviews (
  ReviewID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  Rating INT(5) NOT NULL,
  Comment VARCHAR(255) NOT NULL,
  UserID INT,
  FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE,
  ProductID INT,
  FOREIGN KEY (ProductID) REFERENCES Products(ProductID) ON DELETE
CASCADE
);
```

10. **Table CartItems**:

```
CREATE TABLE CartItems (
  CartItemID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  ProductID INT,
  FOREIGN KEY (ProductID) REFERENCES Products(ProductID) ON DELETE
CASCADE,
  Quantity INT NOT NULL,
  UserID INT,
  FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE
);
```

11. **Table Orders**:

```
CREATE TABLE Orders (
  OrderID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  OrderDate DATETIME NOT NULL,
  TotalPrice INT(255) NOT NULL,
  StatusID INT,
  FOREIGN KEY (StatusID) REFERENCES Status(StatusID) ON DELETE CASCADE,
  UserID INT,
  FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE
);
```

12. **Table Notifications**:

```
CREATE TABLE Notifications (
  NotificationID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  Title VARCHAR(30) NOT NULL,
  Message VARCHAR(255) NOT NULL,
  SendDate DATETIME NOT NULL,
  UserID INT,
  FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE,
  StatusID INT,
  FOREIGN KEY (StatusID) REFERENCES Status(StatusID) ON DELETE CASCADE,
  Type VARCHAR(55) NOT NULL
);
```

13. **Table Invoices**:

```
CREATE TABLE Invoices (
  InvoiceID INT AUTO_INCREMENT PRIMARY KEY,
  OrderID INT,
  FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE,
  InvoiceDate DATETIME NOT NULL,
  Amount DECIMAL(10, 2) NOT NULL,
  BillingAddress VARCHAR(255) NOT NULL,
  StatusID INT,
  FOREIGN KEY (StatusID) REFERENCES Status(StatusID) ON DELETE CASCADE
);
```

14. **Table Payments**:

```
CREATE TABLE Payments (
  PaymentID INT AUTO_INCREMENT PRIMARY KEY,
  PaymentDate DATETIME NOT NULL,
  Amount DECIMAL(10, 2),
  PaymentMethod VARCHAR(255),
  OrderID INT,
  FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE
);
```

15. **Table PromotedProducts**:

```
CREATE TABLE PromotedProducts (
  PromotedProductID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  ProductID INT,
  FOREIGN KEY (ProductID) REFERENCES Products(ProductID) ON DELETE
CASCADE,
  PromotionID INT,
  FOREIGN KEY (PromotionID) REFERENCES Promotions(PromotionID) ON DELETE
CASCADE,
  StartDate DATETIME NOT NULL,
  EndDate DATETIME NOT NULL
);
```

16. **Table OrderItems**:

```
CREATE TABLE OrderItems (
  OrderItemID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  OrderID INT,
  FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE,
  ProductID INT,
  FOREIGN KEY (ProductID) REFERENCES Products(ProductID) ON DELETE
CASCADE,
  Quantity INT(255) NOT NULL,
  Price INT(255) NOT NULL,
  Subtotal INT(255) NOT NULL
);
```

Our team has integrated the Jira with the Github for storing code and collaborating with others easier. We can track the commits from the Jira issue.

Link to github: https://github.com/trinhquykhang/COS20031

Team Health Monitor

Team name	**COS20031-G05**
Sponsor	Swinburne University of Technology
Health monitor cadence	• **Monthly check-in:** This is a quick and easy method to assess how your team members are feeling. You can utilize a poll, a town hall meeting, or one-on-one interactions to do this. • **Quarterly review:** This is a more thorough examination of your team's health. This is a good moment to talk about your team's goals, successes, and issues. You may also utilize this opportunity to identify any areas in which your team requires assistance. • **Annual review:** This is a thorough examination of your team's health. This is an excellent opportunity to discuss your team's performance, triumphs, and lessons learned. This is also a good opportunity to create goals for the following year.

Team health assessment

With your team, read the definition of each attribute of healthy, high-performing teams out loud. On the count of three have each person rate how they feel the team is doing compared to each definition (thumbs-up/green, thumbs-sideways/yellow, thumbs-down/red). Record the results of each attribute rating in the table. Highlight each cell using this color code: HEALTHY= "We're strong here", BIT SICK = "We're ok... but a little shaky", SICK = "We're not healthy".

Full-time owner

There is **one lead who is accountable** for the result of this project. This needs to be someone whose time is at least 80% dedicated to it, and who can champion the mission inside and outside of the team.

9 Sept 2023	Duy Khang took on the role of our full-time owner, but due to some personal issues, he was unable to devote 80% of his time to the project. We understand his situation and are exploring ways to redistribute responsibilities and resources to compensate for his reduced availability, which should be fine, but we are feeling a bit worried. sedimentation at this stage.
16 Sept 2023	Duy Khang has not yet completed his personal work, so he cannot devote all his time to the project. Although it is not possible to devote 80% of the time to this project, with confidence in Duy Khang's management ability in previous times and the team's automation ability. We believe that my project will develop strongly.
23 Sept 2023	He willingly expanded his working hours so that he could devote at least 80% of his time to this project in order to aid in its development and successful completion. Strong and productive working conditions are being produced by Duy Khang's dedication to and focus on the project.
30 Sept 2023	Duy Khang has made this project his top priority and has created a work schedule that will allow him to give it at least 80% of his attention. The project was significantly intensified thanks to Duy Khang's excellent time management and clear vision.
9 Oct 2023	Duy Khang has tried to apply necessary measures to ensure efficiency in time and resource management. By spending more than 80% of his time on the project, Duy Khang is increasingly bringing clear progress to the project.
16 Oct 2023	Duy Khang encountered some difficulties in project management during this time. There were some problems that caused the project progress to decrease, so we felt a bit worried during this time.

⚖️ Balanced team

Roles and responsibilities are clear and agreed upon. The project has people with the right blend of skill set. Acknowledge that team members can change by stage.

9 Sept 2023	In the first stages we divided each task appropriately for each team member, but we were at a disadvantage when some problems appeared unexpectedly causing many problems and damage to our project.
16 Sept 2023	We already know how to set up and divide tasks effectively, but because we are in the process of getting used to new tasks, when unusual tasks arise, we need time to adjust and decide who to do. should take on responsibility. This can cause problems and reduce project implementation time.
23 Sept 2023	Since we are used to taking pauses from our responsibilities, we have procedures and plans in place to deal with any unique scenarios that may emerge. By doing so, we have been able to maintain productivity and guarantee steady project advancement.
30 Sept 2023	We have shared our knowledge and skills in order for everyone to complete their jobs as soon as feasible and to accomplish the project in an efficient and timely manner.
9 Oct 2023	We have clearly identified who is responsible for what aspects of the task, and we have procedures and strategies in place to deal with any unique circumstances that may occur. This was quite helpful to us as we implemented our project.

16 Oct 2023	We have divided responsibilities very clearly and scientifically, and team members have a strong grasp of their assigned tasks. The consensus and willingness to support each other among team members ensure that the project progresses on schedule. This enables us to easily navigate unexpected situations as we are always prepared to respond and find quick solutions.

💞Share understanding

The team has a **common understanding of why they're here**, the problem/need, are convinced about the idea, confident they have what they need, and trust each other.

9 Sept 2023	We have clearly identified who is responsible for what aspects of the task, and we have procedures and strategies in place to deal with any unique circumstances that may occur. This was quite helpful to us as we implemented our project.
16 Sept 2023	Increase trust and support among team members to enhance the spirit of change. Mutual understanding and mutual trust are the keys to creating high team performance.
23 Sept 2023	Project diagrams are created by the team leader to communicate project tasks and goals so everyone can visualize how everything connects together.
30 Sept 2023	Encourage team members to offer suggestions for project solutions or to change strategy as needed. The team is more confidence while adjusting to changes when everyone is clear on the objectives

9 Oct 2023	Hold meetings to ensure that everyone on the team is staying on track with project goals and decisions.
16 Oct 2023	Encourage team members to offer suggestions for project solutions or to change strategy as needed. The team is more confident while adjusting to changes when everyone is clear on the objectives.

💍Value and metrics

It's clear **what success means** from a business and user's perspective, and there is a unique value proposition in place for the target users and for the business. Success is defined, with a goal, and how it will be measured.

9 Sept 2023	Set clear goals for project success. Goals should be derived and built based on the user and business levels.
16 Sept 2023	Determine precise metrics to gauge success. This could include user count, revenue, conversion rates, or other project-related indicators.
23 Sept 2023	We build and develop unique values for both users and businesses. Highlight the project's success in the B2B2C market and create attractive appeal to users.
30 Sept 2023	To grow and improve the project, we assemble opinions and reviews from clients and enterprises. Reviews, direct customer surveys, and the use of social networks can all be used to support this. In order to obtain their perspectives, associations create comment pieces or debates.

9 Oct 2023	We bring the values of products or services and bring them closer to customers clearly and effectively. We can do this using marketing campaigns or social media.
16 Oct 2023	We carry out assessments to check on the successful progress of the project and make necessary adjustments to ensure that everything is going according to plan and achieving goals.

👓Proof of concept

Some sort of demonstration has been created and tested, that demonstrates why this problem needs to be solved, and demonstrates its value.

9 Sept 2023	Proof of concept needs to be created early in the development process to avoid wasting time and resources.
16 Sept 2023	Proof of concept needs to be created by a cross-functional team to ensure that it meets the needs of all stakeholders.
23 Sept 2023	Our team needs a thorough testing process to ensure that the proof of concept works correctly and meets the requirements.
30 Sept 2023	We need to share proof of concept with stakeholders and gather their feedback.
9 Oct 2023	We need to have a process for regularly updating the proof of concept to reflect changes in the solution.

16 Oct 2023	We use proof of concept to create a detailed development plan.

📄One-pager

The **project is summarized** in a one-pager and shared with anyone so that they understand the purpose of the project, and its value.

9 Sept 2023	Our team needs to have a regular one-pager update process, such as: updating one-pager after each important change in the project, updating one-pager before each meeting with stakeholders.
16 Sept 2023	We need to pay attention to one-pager presentations, such as using simple and easy-to-understand language, using images and charts to illustrate, and using titles and subtitles to divide content.
23 Sept 2023	One-pager needs to be shared with the right audience.
30 Sept 2023	One-pager needs to be used effectively to communicate information about the project. Therefore, our team needs to use one-pager in meetings, conferences, and other events to communicate information about the project.
9 Oct 2023	We should regularly review the one-pager to determine if any updates need to be made to reflect changes in the team's perception of the project.
16 Oct 2023	We need to use one-pager to track progress on project goals and targets.

🧘Managed dependencies

Clear understanding of complexity, infrastructure involved, risks, resources, effort, and timeline.

Clear understanding of who we depend on, and who depends on us.

9 Sept 2023	Our team needs to have a clear process for managing dependencies that is appropriate to the scale and nature of the project. This process needs to clearly identify the steps to be taken, the parties involved and the responsibilities of each party.
16 Sept 2023	Our team needs tools and technology to support dependency management. Therefore, the team needs to learn and use appropriate tools and technologies to support dependency management. These tools can help our team track the progress of dependent tasks, get notified about changes, and detect potential issues.
23 Sept 2023	We need to identify and evaluate risks due to dependence, and at the same time put in place measures to minimize these risks such as: diversifying supply sources, building backup solutions, and enhancing adaptability.
30 Sept 2023	The team needs to develop a plan to test and evaluate dependency management. This plan should identify the evaluation criteria and frequency of evaluation. This will help the team ensure that dependency management is being done effectively.
9 Oct 2023	Dependency management is an important skill for teams. We need to train and develop dependency management skills for team members. Therefore, the team needs to build a dependency management training program for team members. This training should cover: basic concepts of dependency management, dependency management skills and tools, dependency management best practices
16 Oct 2023	Dependency management is an important part of project management. We need to integrate dependency management into their project management

	process. We need to identify project dependencies and develop an appropriate dependency management plan. This plan needs to be integrated into the team's project management process.

🏇Velocity

The team is making **incremental progress** by shipping concrete iterations to stakeholders (and, even better, to production), learning along the way, and **implementing lessons learned**, resulting in greater success.

9 Sept 2023	The team's growth rate is at a high level, but there is still room for further improvement. Therefore, the team needs to have a more detailed and long-term development plan to ensure a stable and sustainable growth rate.
16 Sept 2023	Our team needs to have specific metrics to evaluate development speed. This will help the team track and improve development more effectively. So that the team needs to determine specific metrics to evaluate development speed, such as: number of features released, project completion time, ...
23 Sept 2023	Our team needs close coordination among members. This will help the team develop at a faster pace. Therefore, the team needs to have activities to enhance coordination among members, such as: regular meetings, knowledge sharing sessions, training sessions, ...
30 Sept 2023	We need to seek support from external resources, such as consulting companies, experts, communities, ... This will help the team develop at a fast pace.
9 Oct 2023	The team needs to regularly evaluate factors such as: knowledge and skills of team members, tools and technology used by the team, and the team's

	working environment. Then, coming up with measures to improve them.
16 Oct 2023	The team's growth needs to be balanced with the quality of the product or service. The team needs to have a reasonable plan to develop the product or service, ensuring both speed and quality.

🎯 Focus areas

Ask your team to collectively come up with one attribute you want to focus on. Then, call out ways to move the SICK or BIT SICK toward HEALTH. Make sure they are actionable, specific, and measurable.

Date	Focus areas and action items
Sep 10, 2023	□ Determine constraints and assess the system's present scalability.
Sep 10, 2023	□ Assign reasonable tasks to team members so they can cooperate to address problems when the full-time owner is unable to do so.
Sep 19, 2023	□ Create backup plans in advance so that you can quickly and effectively address unforeseen issues.
Sep 25, 2023	□ To prevent the project from being too significantly impacted by managing uncommon jobs, concentrate on implementation and identify priority activities first.
Oct 9, 2023	□ Perform security testing and vulnerability assessments from the system
Oct 18, 2023	□ Use performance monitoring tools to implement regular system performance analysis.

🌹 Next step

1. ER diagrams, database diagrams, database normalization, and MySQL installation and database creation are important steps in the process of developing an information system.
2. ER diagrams are used to define the data requirements of the system. ER diagrams can be used to describe entities, attributes, and relationships between entities in a system.
3. The database diagram is created from the ER diagram. Database diagram describes the structure of the database. A database diagram includes tables, primary keys, and foreign keys.
4. Database normalization is the process of organizing data in a database to minimize data redundancy and improve data integrity. Database normalization can be performed in five levels, from level 1 to level 5.
5. Installing MySQL and creating the database is the final step in the database development process. MySQL is an open source relational database management system (RDBMS). MySQL is widely used in web applications and mobile applications.