4/8/2024

<PAYSTUB+> SYSTEM REQUIREMENTS DOCUMENT
DRAFT

Paystubbers Inc.

Michael B, Ethan R, Daniel G, Jimmy P

Table of Contents

INTRODUCTION (<MB>)

Describe the purpose of this requirements document and outline what it contains.

The purpose of this document is to describe the functionality of the app and the design of it as well. In the description model, it'll describe how each aspect of the app will work when the users use it. For example, when a user puts in their income and tips, there's a process for that. In the class diagram section, it will contain and display what classes will be made in the system and what functions and attributes in the respective classes. In the use case diagram, it will show how the users will interact with our app, like if the user checks their income for example. In top of that, there is an explanation of each case in the use case scenario section which will go into detail on what happens in each case. The system sequence charts section will contain graphs that will display how the actions will play out when the user uses the features on the app.

DESCRIPTION MODEL (<ER>)

Using text, describe the requirements for your system. Expand on the function section from your project plan. Include requirements for the following categories: Output, Input, Processes, Performance and Security.

Output Requirements:
- Display comprehensive earnings data including cash and credit tips, hourly pay, and peak income periods.
- Present weekly, monthly, and annual income summaries.
- Provide graphical representations of income trends through tables and graphs.
- Customizable output based on user preferences for tip-out methods and tax deductions.
- Push notifications for reminders and alerts.
- In-app communication feature for seamless conversations among coworkers.

Input Requirements:
- Ability to input work hours, tips (cash and credit), and tax information.
- Customizable settings for tip-out methods and tax deductions.
- Allow managers to input income and tip data for serving staff.
- Support for tracking multiple jobs simultaneously.

Processes Requirements:
- Organize income records accurately based on user-inputted data.
- Adapt to different tip-out methods (e.g., pooling or individual distributions).
- Calculate state tax percentages and deduct taxes from earnings.
- Send push notifications for timely input of work hours and tips.
- Facilitate in-app communication among coworkers.

- Allow managers to create restaurant profiles and manage income data.

Performance Requirements:
- Provide real-time data updates and calculations.
- Ensure seamless performance even with multiple users and large datasets.
- Responsive user interface for smooth interaction and data entry.
- Minimal latency in sending push notifications and in-app communication.
- Ability to handle concurrent user requests without degradation in performance.

Security Requirements:
- Implement secure user authentication and authorization mechanisms.
- Encrypt sensitive user data (e.g., earnings, tax information).
- Secure transmission of data over the network using encryption protocols.
- Implement role-based access control to restrict access to sensitive features (e.g., manager privileges).
- Regular security audits and updates to address potential vulnerabilities.
- Compliance with relevant data protection regulations (e.g., GDPR, CCPA).

CLASS DIAGRAM (<ER>)

Create a class diagram. The Class Diagram should contain all of the system objects, their attributes, and any known methods. This diagram may be included as a separate file – it does not need to be inserted into this Word document.

USE CASE DIAGRAM(<JP>)

Create a Use Case Diagram for all of the "uses" of your system.   This diagram may be included as a separate file – it does not need to be inserted into this Word document.

USE CASE SCENARIOS(<MB>)

Create a full description Use Case Scenario (detailed descriptions) for each use case of the system. This full scenario should include an enumerated list of steps involved in the activity as well as any exception conditions.

SYSTEM SEQUENCE CHARTS(DANNY)

For each Use Case Scenario, provide a sequence diagram. Use your class diagram, use case diagram and scenarios to create the corresponding System Sequence Diagram.