**BOOKING FLIGHT SYSTEM**

**GROUP 10**

**SYSTEM DESIGN DOCUMENT**

***Overview***

*Hệ thống quản lý bán vé máy bay được thiết kế dựa trên nhu cầu của đồ án môn học, các yêu cầu của nhóm đối tác và nhu cầu thực tế của người dùng. Hệ thống được thiết kế để hoạt động trên môi trường điện thoại smartphone thân thiện với người dùng. Hệ thống được thiết kế với các chức năng cơ bản như đặt vé, chỉnh sửa thông tin cá nhân, đăng nhập tài khoản, thêm các dịch vụ, thanh toán, check in,…*

*The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, and external interfaces.*

**I.INTRODUCTION**

1. **Purpose and Scope**

Mục đích của đồ án này là thiết kế ra một hệ thống bán vé máy bay dựa trên yêu cầu từ giảng viên kết hợp với yêu cầu từ nhóm đối tác. Đồng thời dựa trên những trải nghiệm thực tế để cái tiến, nâng cao trải nghiệm người dùng. Mong muốn của nhóm là tạo ra một hệ thống có thể sử dụng rộng rãi và hiệu quả với mọi người sử dụng.

1. **Project Executive Summary**

**2.1. System Requirement**

a.Functional

RQ1: Tài khoản cho khách hàng

RQ2: Tài khoản cho nhân viên

RQ3: Truy cập, cập nhật CSDL quản lí vé

RQ4: Quản lí thông tin đại lí máy bay

RQ5: Dịch vụ tư vấn khách hàng

RQ6: Hình thức thanh toán( trả trước, tiền mặt…)

RQ7: Đại lí có thể tổng kết từ từng chuyến bay( doanh thu, số vé bán, tỉ lệ bỏ vé….)

RQ8: Tài khoản khách hàng có quy định về việc đặt vé (trước 1 ngày) và quy định về đổi trả vé

RQ9: giao diện đặt vé, hiển thị chuyến bay( sort theo hạng vé, giá, ngày giờ)

RQ bên customer: None

**2.2 System Overview**

Hệ thống này được xây dựng dựa trên nhu cầu về việc đặt vé online cũng như việc quản lý dữ liệu đang ngày càng được số hoá. Việc xây dựng một ứng dụng không còn quá phức tạp. Qua đó, nhóm quyết định tạo ra một ứng dụng với quy mô nhỏ, quản lý dữ liệu trên một database. Cùng với đó sẽ thiết kế một giao diện thân thiện với người dùng, một cơ chế xử lý dữ liệu hiệu quả và nhanh chóng. Với đồ án môn học này, nhóm mong muốn sẽ có thể học hỏi thêm nhiều kỹ năng cũng như làm quen với môi trường làm việc cho sau này.

1. **Document Organization**

Nhóm phát triển:

**21520769 - Hồ Đình Duy**

**21520894 - Nguyễn Duy Hưng**

**21520905 - Nguyễn Thị Thu Hương**

**21521898 - Bùi Cường**

Nhóm khách hàng: **Nhóm 9**

1. **Glossary**

Thuật ngữ và từ viết tắt

UI - User interface (Giao diện người dùng)

ERD - Entity Relationship Diagram

MySQL – local database

**II. SYSTEM ARCHITECTURE**

Hệ thống sẽ tiếp nhận thông tin, yêu cầu từ khách hàng. Tuỳ thuộc vào yêu cầu của khách hàng như mua vé, thêm dịch vụ hay để từ giao diện, ta thực hiện ánh xạ các thao tác của khách hàng thành những hành động, tương tác tác động lên database. Từ đó thay đổi, lưu trữ lại những dữ liệu, những hành động đó.

**1.System Hardware Architecture**

Hệ thống sử dụng máy tính cá nhân của sinh viên để lưu trữ source code và điện thoại di động cá nhân của sinh viên để thử nghiệm.

**2.System Software Architecture**

Tổ chức phần mềm tổng thể:

* Xác định những chức năng theo yêu cầu:
* Dựa trên yêu cầu đề bài.
* Dựa trên yêu cầu của nhóm đối tác.
* Dựa trên nhu cầu thực tế sử dụng.
* Làm rõ các chức năng chi tiết trong từng màn hình.
* Làm rõ và mô hình hoá các chức năng dưới dạng sơ đồ.
* Thiết kế UI:
* Xây dựng các mô hình các bản vẽ UI dự kiến
* Kiểm duyệt, chọn lọc và khảo sát ý kiến người dùng
* Xây dựng các mô hình UI phục vụ trực tiếp cho ứng dụng.
* Xây dựng một mô hình dữ liệu để lưu trữ các thông tin của hệ thống.
* Thực hiện lưu trữ dữ liệu local với MySQL.
* Xây dựng các chức năng cho hệ thống:
* Xây dựng chức năng cho từng màn hình.
* Xây dựng source code các chức năng dựa trên các mô hình đã xây dựng trước đó.
* Tích hợp và tiến hành kiểm duyệt.
* Thực hiện tích hợp và kiểm duyệt các chức năng của hệ thống.
* Thực hiện giai đoạn thử nghiệm của ứng dụng.

**III. FILE AND DATABASE DESIGN**

Các dữ liệu trong hệ thống cần quản lý bao gồm:

* Tài khoản của khách hàng
  + Tài khoản
  + Mật khẩu
  + Thông tin cá nhân: Ảnh đại diện, tên, tuổi ,địa chỉ, số điện thoại, email, giới tính, quốc tịch.
  + Các vé đã đặt
  + Các vé đã từng đặt
* Tài khoản admin
  + Tài khoản
  + Mật khẩu
  + Lịch sử các chỉnh sửa dữ liệu
    - Thêm
    - Xóa
    - Chỉnh sửa chuyến bay
  + Quyền hạn
* Dữ liệu các chuyến bay
  + Dữ liệu các chuyến bay
    - Tên chuyến bay
    - Điểm xuất phát
    - Điểm đến
    - Thời gian cất cánh
    - Thời gian bay
    - Thời gian hạ cánh dự kiến
    - Các hạng ghế: thương gia, phổ thông

Dữ liệu các vé

* + - Mã vé
    - Mã chuyến bay
    - Hạng vé
    - Nơi đi, nơi đến
    - Ngày khởi hành

1. **Database Management System Files**

* Các dữ liệu của hệ thống sẽ được lưu trữ trên một database MySQL.
  + Với việc dữ liệu được lưu trữ local thì ứng dụng có thể xem như hoạt động liên tục không bị gián đoạn.
  + Dễ dàng quản lý ở mọi nơi, mọi lúc.
  + Việc thêm, chỉnh sửa có thể được thực hiện nhanh chóng do dữ liệu local.
  + Hiệu suất truy xuất cao do dữ liệu lưu trữ local
* Các thông tin sẽ được tổ chức dưới dạng bảng để lưu trữ. Độ chi tiết của dữ liệu truy xuất sẽ được lựa chọn.

**IV. HUMAN-MACHINE INTERFACE**

1. **Inputs**

Hệ thống này sẽ nhận đầu vào dựa trên tương tác của người dùng với giao diện (ở đây cụ thể là màn hình điện thoại)

* Các dữ liệu có thể thu thập được từ người dùng bao gồm:
  + - * Tài khoản, mật khẩu
      * Thông tin cá nhân như tên, địa chỉ, số điện thoại, email, giới tính
      * Các thông tin mua vé, dịch vụ đăng ký
      * Yêu cầu liên hệ CSKH
* Các thông tin thu được từ tài khoản admin:
  + - * Các thông tin đăng nhập của tài khoản admin: Tài khoản, mật khẩu
      * Các thông tin cá nhân của tài khoản admin
      * Các tương tác thêm, xóa, sửa database

1. **Outputs**

* Đối với khách hàng:

Đầu ra sẽ bao gồm

Các giao diện tương tác với người dùng

Các vé, các dịch vụ được sở hữu, mua bởi khách hàng

Các vé hiển thị sau khi lọc dưới một số yêu cầu của khách hàng

Dịch vụ CSKH được kết nối

* Đối với admin:  
   Đầu ra sẽ bao gồm:

Các giao diện tương tác với admin.  
 Các chuyến bay, các vé máy bay được thêm, xóa, sửa khỏi database.

**V. DETAILED DESIGN**

1. **Hardware Detailed Design**

Thiết bị cá nhân được sử dụng trong đồ án:

* Acer Nitro 5:
  + CPU: Intel i5 - 11400H
  + Ram: 8GB
  + OS: Windows 11
  + GPU: RTX-3050
* Samsung Galaxy A53 5G
  + Android 13

1. **Software Detailed Design**

* Xác định những chức năng theo yêu cầu:
* Từ những yêu cầu, ta xác định các chức năng chính của hệ thống
* Tinh chỉnh và hệ thống lại các chức năng dưới sơ đồ use-case
* Phân tích các yêu cầu của người dùng, xác định những thuộc tính, các dữ liệu cơ bản mà ta cần phải quản lý:
  + Xây dựng các mô hình ERD.
* Làm rõ và mô hình hoá các chức năng dưới dạng sơ đồ.
  + Lập ra báo cáo phân tích các chức năng, nhu cầu của người dùng.
  + Xây dựng Class Diagram xác định các hàm, thông tin quản lý và chức năng.
  + Trừu tượng hoá giải thuật code thông qua activity diagram.
  + Xây dựng tương tác giữa các hệ thống bằng sequence diagram.
* Thiết kế UI:
* Xây dựng các mô hình các bản vẽ UI dự kiến
  + Sử dụng công cụ figma để xây dựng nhanh chóng các mô hình UI mô phỏng.
  + Dựa trên các tài hệ thống sẵn có và yêu cầu người dùng.
  + Dựa trên các quy tắc về thiết kế giao diện người dùng dễ nhìn, thu hút.
* Kiểm duyệt, chọn lọc và khảo sát ý kiến người dùng
  + Thực hiện khảo sát, hỏi ý kiến khách hàng và các nhóm khác về mức độ hài lòng của UI mô phỏng
  + Thực hiện sàng lọc và tinh chỉnh lại các hệ thống UI chưa đáp ứng yêu cầu
* Xây dựng các mô hình UI phục vụ trực tiếp cho ứng dụng.
  + Sử dụng công cụ Android Studio để thiết kế lại các UI đã qua kiểm duyệt
  + Thiết kế giao diện bằng Layout Editor
  + Thực hiện chạy trên điện thoại cá nhân để xem xét, chỉnh sửa
* Xây dựng một mô hình dữ liệu để lưu trữ các thông tin của hệ thống.
* Thực hiện lưu trữ dữ liệu ở Local Database:
  + Xây dựng một dự án trên MySQL.
  + Lựa chọn xây dựng một local-database trên dự án đó.
  + Tích hợp, kết nối database đã tạo ở trên cho hệ thống.
* Xây dựng các chức năng cho hệ thống:
* Xây dựng chức năng cho từng màn hình.
  + Dựa trên các chức năng đã phân tích ở báo cáo phân tích chức năng, nhu cầu người dùng. Từ đó xác định các chức năng, các tương tác xuất hiện trong từng màn hình.
  + Dựa trên sơ đồ lớp để xác định các thông tin, thuộc tính, các thao tác trên các lớp đối tượng.
  + Xây dựng các lớp đối tượng. Xây dựng các hàm, các hành động, các thao tác trên lớp đối tượng.
* Hoàn thiện các source code, các hàm, các thao tác đã xây dựng.
* Tích hợp và tiến hành kiểm duyệt.
* Thực hiện tích hợp hệ thống.
  + Tích hợp các thao tác đã xây dựng với hệ thống UI đã xây dựng trước đó
  + Tích hợp các chức năng với hệ thống MySQL database đã tạo ở trước đó
  + Thực hiện kiểm thử mỗi khi tích hợp 1 hệ thống nhỏ để dễ dàng phát hiện lỗi và giảm chi phí, sức lực sửa lỗi.
  + Sửa lại các lỗi xuất hiện
  + Khảo sát ý kiến khách hàng, nhóm đối tác để chỉnh sửa lại theo yêu cầu
* Thực hiện giai đoạn thử nghiệm của ứng dụng.
  + Hoàn thiện ứng dụng và tiến hành thử nghiệm
  + Lập khảo sát về trải nghiệm người dùng
  + Tổng hợp, báo cáo, bảo dưỡng cải tiến hệ thống.

**VI. EXTERNAL INTERFACES**

1. **Interface Detailed Design**

Với hệ thống bán vé máy bay ta cũng cần các quy tắc về thiết kế giao diện riêng. Các thông tin chi tiết về giao diện sẽ được ghi dưới đây:

* Thiết kế ngôn ngữ:

Định dạng ngôn ngữ cho toàn bộ giao diện: Song ngữ

Phông chữ sử dụng: Montserrat

* Giao diện được thiết kế bằng Android Studio:

Ngôn ngữ lập trình: Java

Định dạng lưu trữ, truyền tải: Java

Nơi lưu trữ: Github

* Công cụ truyền dữ liệu:

Tải trực tiếp qua cổng USB Type C hoặc qua mạng wifi (2 thiết bị cùng mạng)

* Báo lỗi:

Tự động báo trực tiếp về server

* Các bước thực hiện khi phát hiện lỗi:
* Lỗi được phát hiện và báo cáo về cho admin
* Chỉ định lập trình viên sửa lỗi
* Kiểm tra lại lỗi đã được sửa hay chưa, có phát sinh lỗi mới hay không, viết báo cáo lỗi
* Lỗi sẽ được sửa trong bản cập nhật kế tiếp

**VII. SYSTEM INTEGRITY CONTROLS**

* Hệ thống được thiết kế để tránh một vài trường hợp mất kiểm soát hệ thống như:
  + Tránh rò rỉ các thông tin cá nhân: Bằng cách sử dụng hệ thống phân quyền cũng như hầu hết các thông tin cá nhân hầu như không thể truy cập từ bất kì tài khoản nào khác; các thông tin của người dùng gần như được bảo mật tuyệt đối
  + Thiết kế các cơ chế kiểm soát thời điểm chỉnh sửa, truy cập có thể nhanh chóng xác nhận được các lỗi cũng như các hành động nghi ngờ.