Project Name: Housing Supply, Demand and Waiting List Data Application

The lack of effective tools for local governments to manage housing supply and waiting list data leads to difficulties in data analysis and inefficient decision-making. Therefore, designing an efficient housing data management platform can help local governments achieve centralized management, real-time query and data analysis support.

Advantages of this website:

1. Improve data management efficiency.
2. Provide dynamic query and data analysis functions.
3. Provide a user-friendly interface.
4. Ensure data accuracy and reliability.

Beneficiary users of the website:

Local government officials:

1. Check housing supply and waiting lists in different regions.
2. Compare historical data and analyze trends in housing demand.
3. Generate reports based on data to support policy adjustments.

Data analysts:

1. Filter data by time and region to generate visual reports (such as trend charts, bar charts).
2. Predict future housing supply and demand in a region and identify resource allocation priorities.

IT system administrators:

1. Regularly update and maintain regional and annual data.
2. Ensure the normal operation of the system, regularly back up the database, and ensure data integrity.

The public:

1. Understand the housing policy and current supply situation in your area.
2. Check your waiting list ranking and queue progress.
3. View historical housing supply and demand data and trends in the region.

Users can see key information from this website, such as:

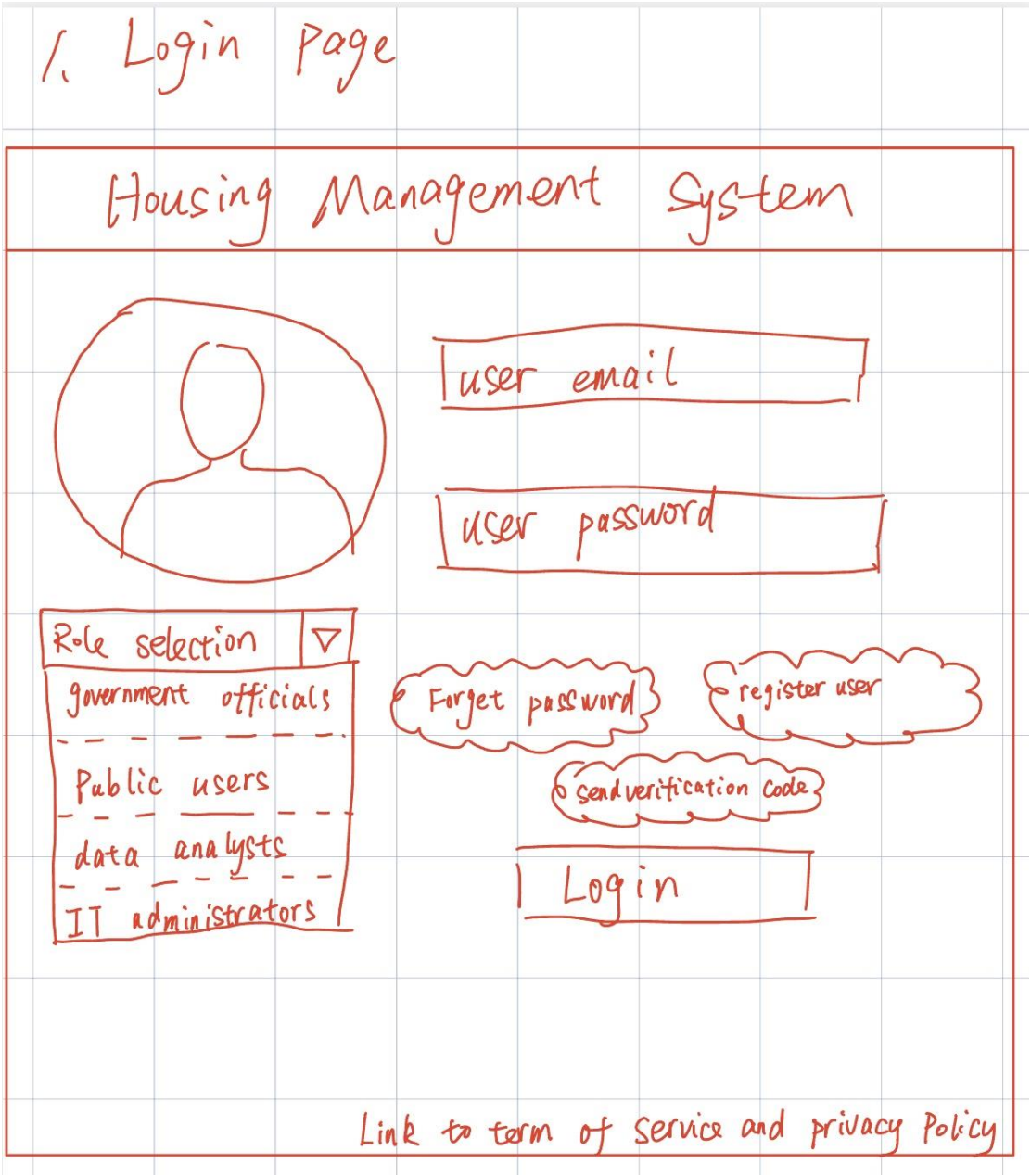
1. What is the current total housing supply in the region?
2. What is the distribution of waiting list numbers in each region?
3. Which regions have the largest supply and demand differences?

Prioritise the user stories

NO.	User Stories	MoSCoW Categorisation
1	As a local government official, I want to filter housing data by region and year so that I can generate regional supply and demand analysis reports.	Must Have
2	As a local government official, I would like to see waiting list statistics so I can determine which areas have the most pressing housing needs.	Must Have
3	As a data analyst, I want to generate supply and	Should Have

	demand trend charts so that I can analyze past trends and provide data support for decision making.	
4	As a data analyst, I want to compare housing demand data across multiple regions so that I can identify resource allocation priorities and optimize allocation plans.	Should Have
5	As a public user, I would like to check my waiting list ranking so that I can know my queue status and estimated allocation time.	Must Have
6	As a public user, I would like to view housing supply and demand data in my area so that I can understand the current housing supply and demand situation.	Should Have
7	As a public user, I would like to be notified of housing policy updates so that I can stay up to date on policy changes that affect me.	Could Have
8	As an IT system administrator, I want to maintain regional and year data so that I can ensure that the data in the system is current and accurate.	Must Have
9	As an IT system administrator, I want to back up my database regularly so that I can quickly restore it in case of data loss.	Must Have
10	As a local government official, I want to export the filtered data report so that I can share it with other departments or leaders for decision analysis.	Should Have

Interface Design



1. Login Page

Functions:

User authentication, select role to log in. Drop-down menu: Local government official, public user, data analyst, IT administrator).

System title "Housing Management System".

User name input box.

Password input box.

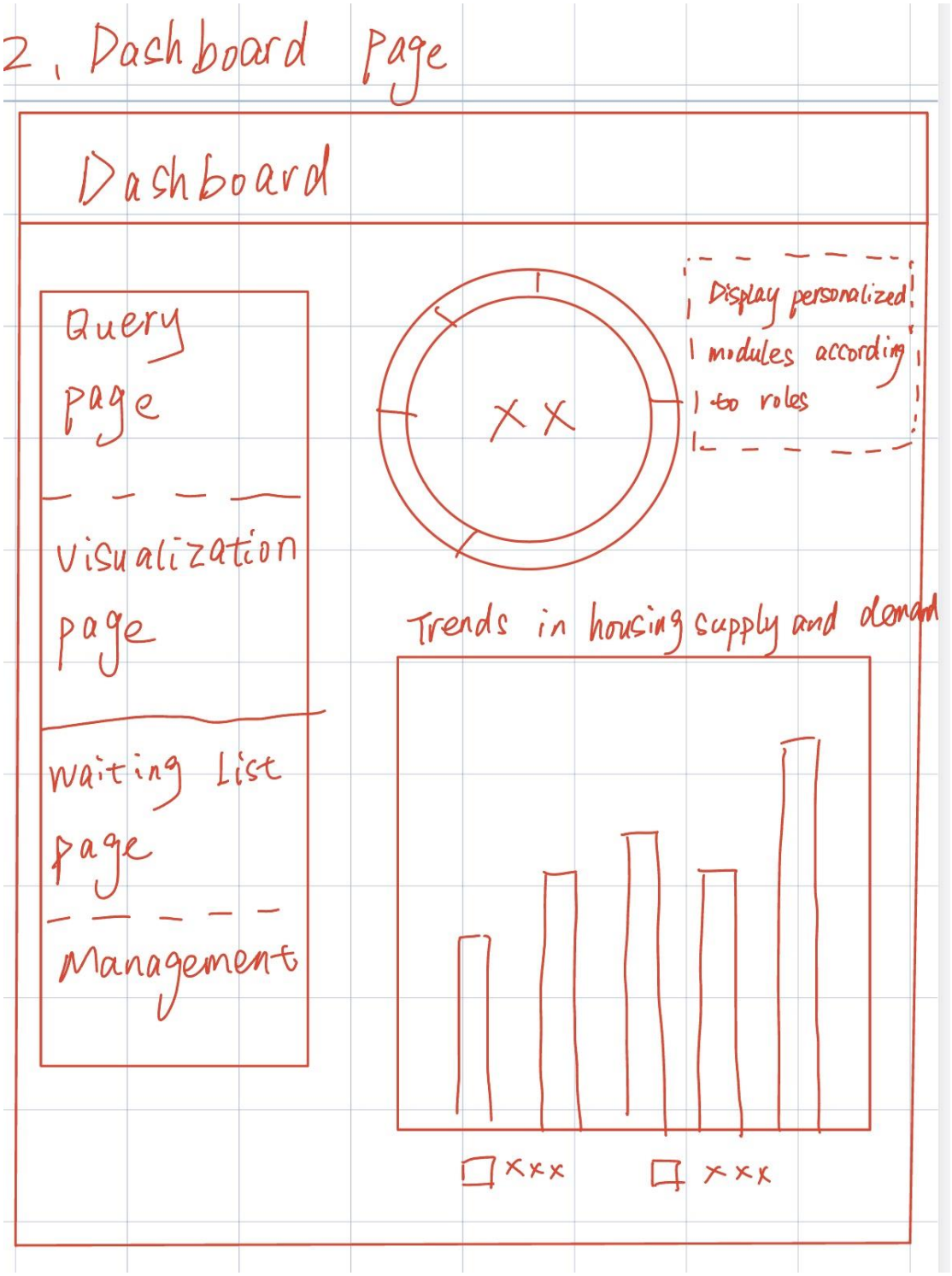
Login button.

Forgot password link

Register

User verification

Privacy permissions

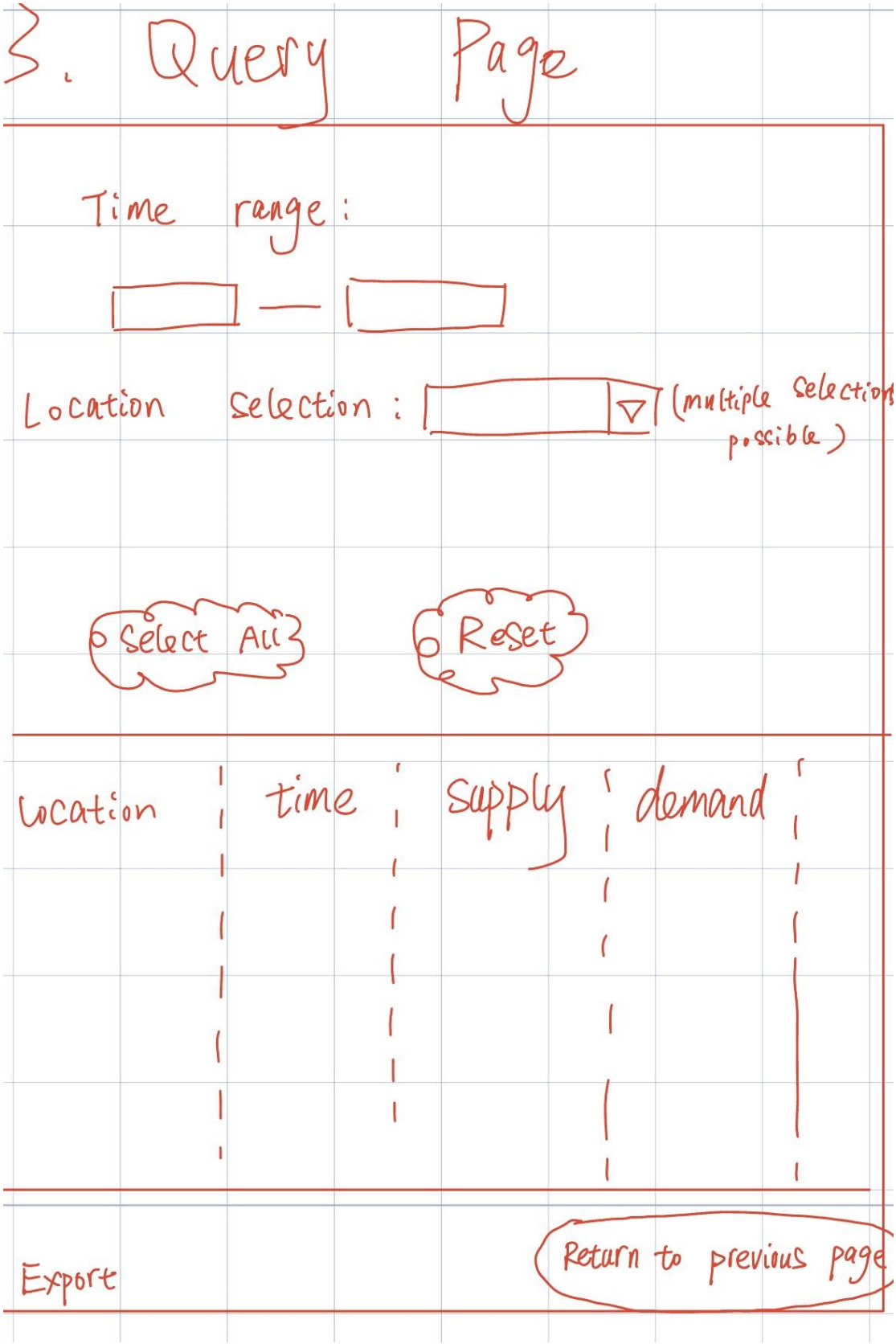


2. Dashboard Page

Function:

Function entry button (query, waiting list, data visualization, data management).

Total housing supply, total demand, total number of people on the waiting list in the current area



3. Query Page

Function:

Filter data and display the filter results.

Select area
Select year
Filter button.
Display the filtered supply and demand data in a table, with columns including: area, year, housing supply, and demand.
Export button (PDF or Excel format).

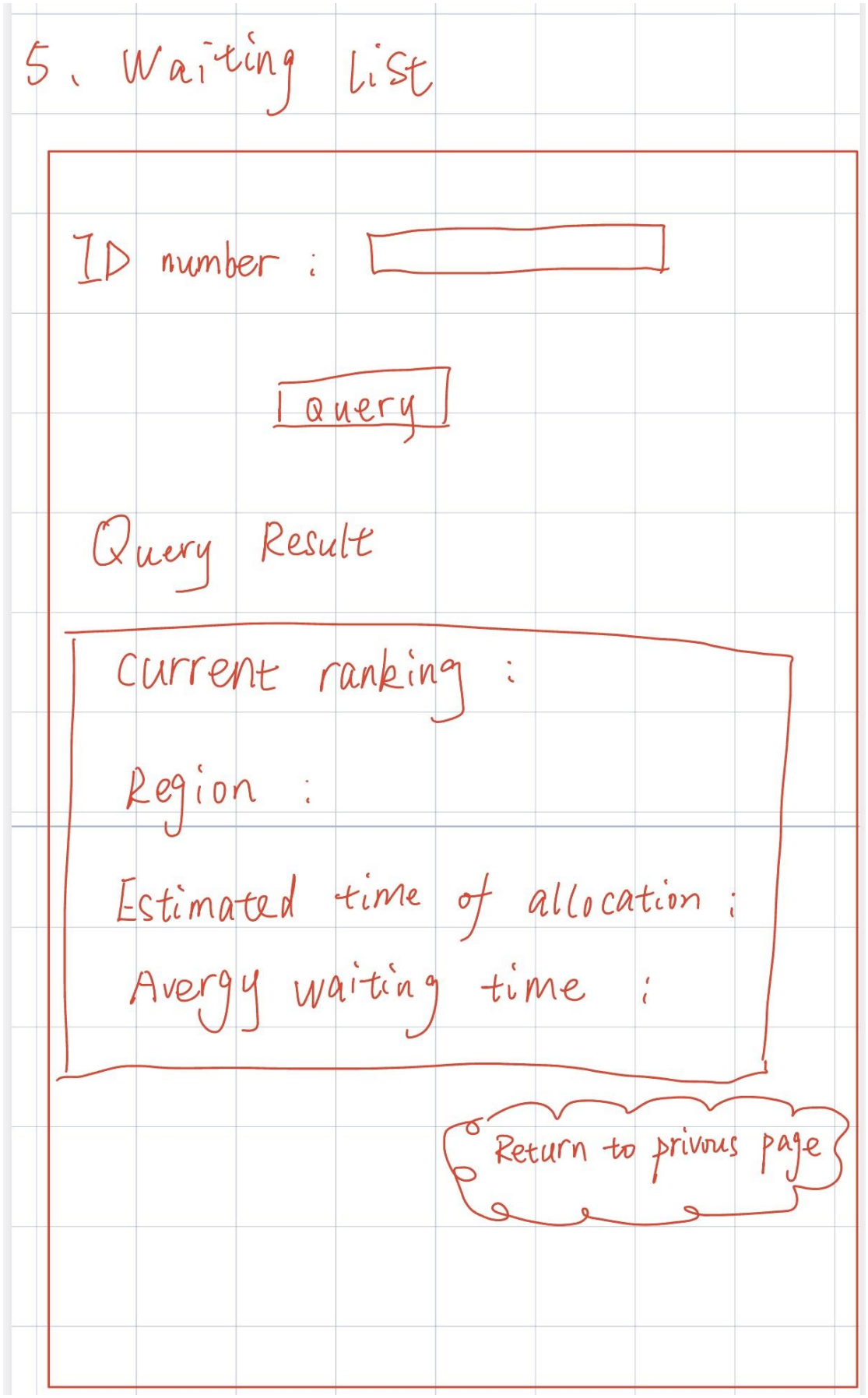
4. Visualization Page

Function:

Display trend charts of housing supply and demand data.

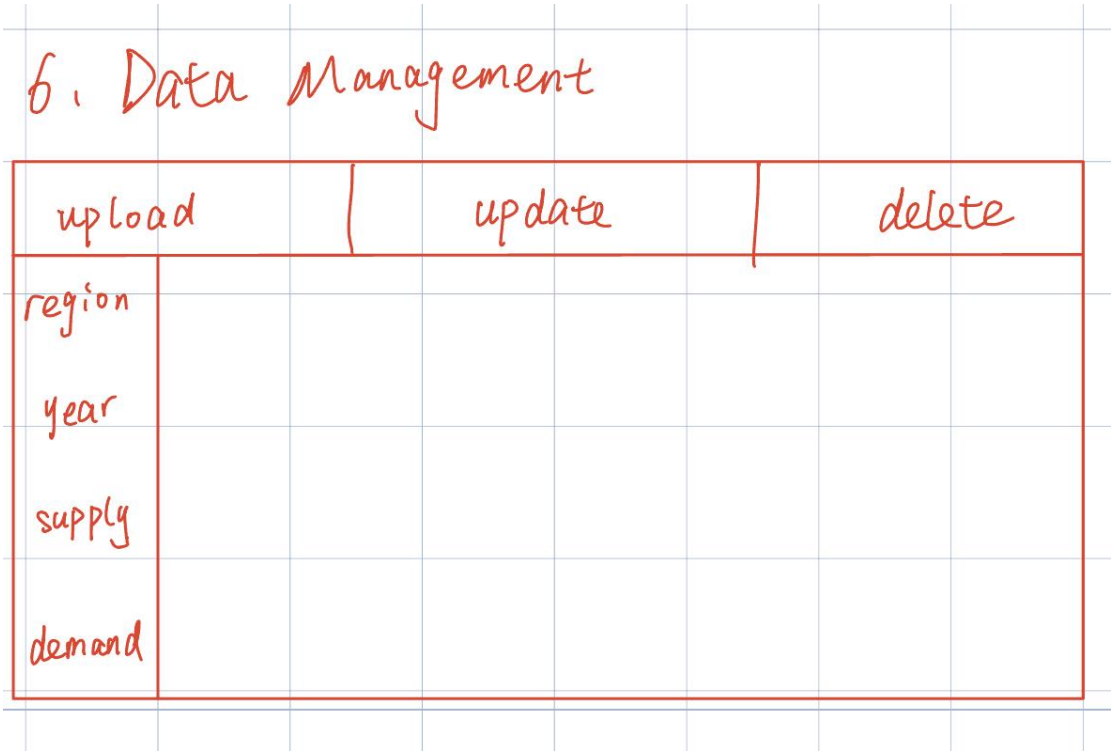
Select data range (area and year).

Select chart type (bar chart, line chart).
Chart display area
Export charts as images or reports.



5. Waiting List Query Page

Function:
Provide public users with information about their waiting lists.
Query area
Enter personal information
Query button.
Query results
Show current ranking and estimated allocation time.
Provide export function



6. Data Management Page
Function:
IT administrator maintains system data.
Layout description:
Data backup and restore buttons.
Data table
Shows regional and year data (columns include: regional name, year, supply and demand data).
Data editing buttons (add, modify, delete).

Application Logic Design

Application logic design
1. Introduction
The core functions of this design: user authentication, housing supply and demand data query,

waiting list management, trend visualization, and administrator data management.

According to the UML diagram, this system is divided into the following key modules:
1. Login Package: handles user login, registration, and password reset functions.
login(username: String, password: String, role: String): boolean: Verify user identity.
registerUser(details: UserDetails): boolean: Register a new user.
resetPassword(email: String): void: Process password reset requests.

2. Authentication Package: Verify user identity and correspond to different permissions according to different roles.
authenticate(username: String, password: String): boolean: Verify username and password.
verifyRole(username: String, requiredRole: String): boolean: Check whether the user role meets the permission requirements.

3. Dashboard Package: Provide users with statistical and announcement information.
getDashboardData(userRole: String): DashboardData: Get relevant dashboard data based on user role.
getSummaryCards(): List: Display quick statistics cards.
getAnnouncements(): List: Get system announcements.

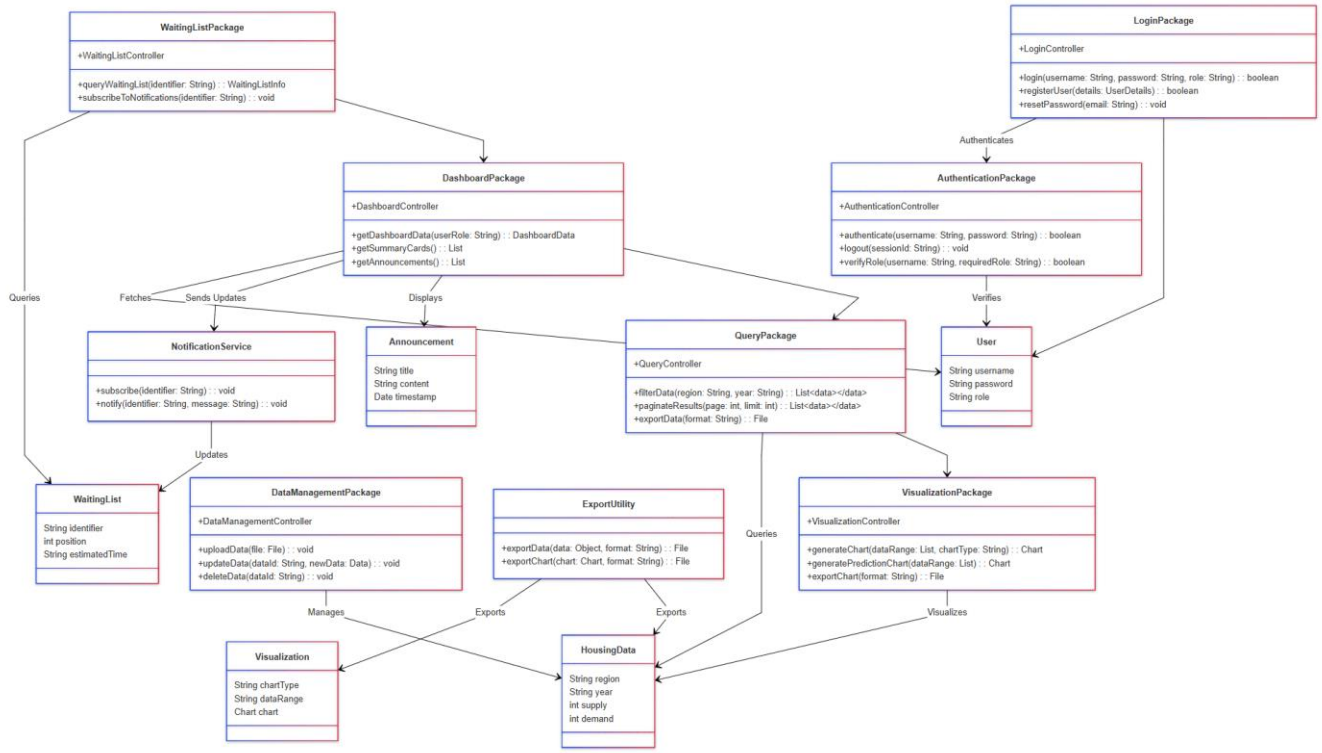
4. Query Module (Query Package): Supports users to filter housing supply and demand data and export results.
filterData(region: String, year: String): List<Data>: Filter data based on region and year.
paginateResults(page: int, limit: int): List<Data>: Display filtered results in pages.
exportData(format: String): File: Export filtered results to PDF or Excel.

5. Visualization Module (Visualization Package): Generate supply and demand trend charts and support export.
generateChart(dataRange: List<String>, chartType: String): Chart: Generate charts (such as bar charts, line charts).
generatePredictionChart(dataRange: List<String>): Chart: Generate prediction trend charts (optional).
exportChart(format: String): File: Export charts as images or reports.

6. Waiting List Package: Manage and query waiting list information for public users.
queryWaitingList(identifier: String): WaitingListInfo: Query user ranking and estimated allocation time.
subscribeToNotifications(identifier: String): void: Subscribe to notifications for waiting lists.

7. Data Management Package: Support administrators to upload, update and maintain housing data.
uploadData(file: File): void: Upload new data files.
updateData(dataId: String, newData: Data): void: Update existing data.
deleteData(dataId: String): void: Delete expired data.

8. Notification Service Package: Send real-time notifications to users.
subscribe(identifier: String): void: Subscribe to user notifications.
notify(identifier: String, message: String): void: Send notification messages to users.



Database Design
Database design

Main parts of the database:
AREA
Stores information about all areas.
area_code (string, PK): area code, primary key.
area_name (string): area name.

YEAR
Stores year information.
year (int, PK): year, primary key.

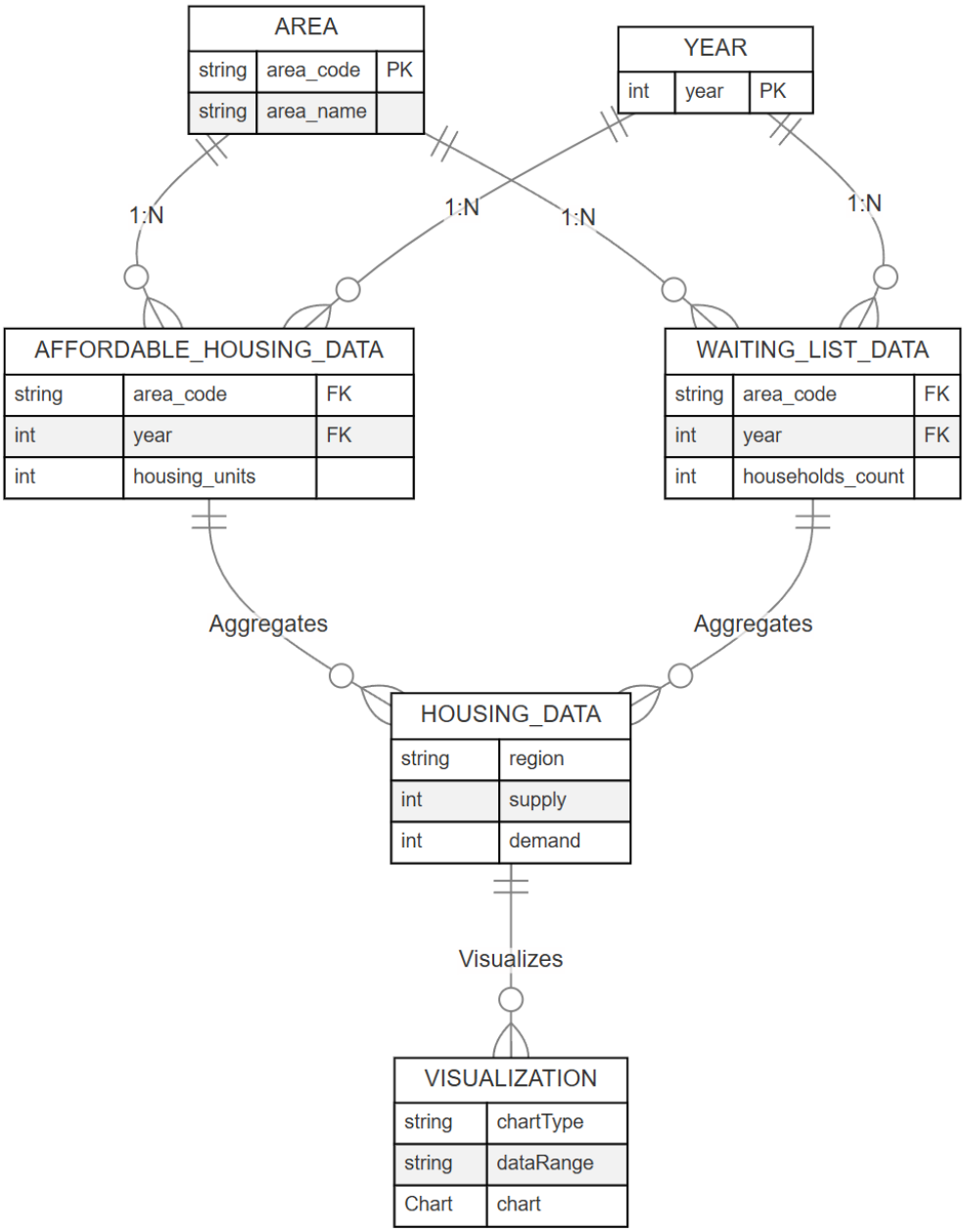
AFFORDABLE_HOUSING_DATA
Records the number of affordable housing units in each area each year.
area_code (string, FK): area code, foreign key, associated with AREA.
year (int, FK): year, foreign key, associated with YEAR.
housing_units (int): number of affordable housing units.

WAITING_LIST_DATA
Records the number of households on the waiting list in each area each year.
area_code (string, FK): area code, foreign key, associated with AREA.
year (int, FK): year, foreign key, associated with YEAR.
households_count (int): number of households on the waiting list.

HOUSING_DATA (housing supply and demand data)
Affordable housing data and waiting list data, used to show the supply and demand relationship.
region (string): region name or code.
supply (int): total housing supply.
demand (int): total housing demand.

VISUALIZATION (visualization)
Stores information for visualizing supply and demand data.
chartType (string): chart type (such as bar chart, line chart).
dataRange (string): data range (such as year or region).
chart (Chart): stores the generated chart object.

Region and affordable housing data: one region (AREA) can correspond to multiple affordable housing records (AFFORDABLE_HOUSING_DATA), and the 1:N relationship is realized through area_code.
Year and affordable housing data: one year (YEAR) can correspond to multiple affordable housing records (AFFORDABLE_HOUSING_DATA), and the 1:N relationship is realized through year.
Area and waiting list data: One area (AREA) can correspond to multiple waiting list records (WAITING_LIST_DATA), and a 1:N relationship is achieved through area_code.
Year and waiting list data: One year (YEAR) can correspond to multiple waiting list records (WAITING_LIST_DATA), and a 1:N relationship is achieved through year.



Reference:

1. I acknowledge using Microsoft Copilot (version GPT-4, Microsoft, https://copilot.microsoft.com/) to build my overall framework and proofread my drafts, and to optimize and rewrite the code.
2. I referred to the tutorial code (https://github.com/152ChenyuYang/comp0035-2024-tutorials.git)