

**2023-2024 FALL SEMESTER**

**CS 491**

**PROJECT SPECIFICATION DOCUMENT**

**SecuRent**

**Supervisor: Özcan ÖZTÜRK**

**Team ID: T2306**

Arda YILDIZ - 22003093

Mustafa Kaan KOÇ - 21903358

Yusuf ŞENYÜZ - 21903105

Cengizhan TERZİOĞLU - 22003014

Yunus Eren TÜRKERİ - 22001842

**Table Of Contents**

[**1. Introduction 2**](#_jekix4dpq2v3)

[1.1 Description 2](#_z9gtbul64fp3)

[1.2 High Level System Architecture & Components of Proposed Solution 2](#_ngr1ce7gjrvx)

[1.2.1 Description of the Architecture Diagram 4](#_d36ybcs8nwrg)

[1.2.2 Stakeholders 5](#_ncd1xninlqfh)

[1.2.3 System Perspectives 5](#_adcny3rtujia)

[1.3 Constraints 6](#_1mp3n3vr1sh9)

[1.3.1 Implementation Constraints 6](#_lw32bc9ycl0n)

[1.3.2 Economic Constraints 6](#_2ywfo2hsqi62)

[1.3.2 Ethical Constraints 7](#_rzvrw933480a)

[1.4 Professional and Ethical Issues 7](#_dtbb6iypfmer)

[1.5 Standards 7](#_7luvxyprpwuf)

[1.5.1 Standards Used 7](#_nhzdr3gvr4v1)

[1.5.2 Standards To Be Used 8](#_p8hpb7pkwc6d)

[**2. Design Requirements 8**](#_f880d1d258c3)

[2.1 Functional Requirements 8](#_htsajedt4gqr)

[2.2 Non-Functional Requirements 9](#_kap2ovhxlt2v)

[2.2.1 Usability 9](#_w9fy5cjj3cob)

[2.2.2 Reliability 9](#_8pvnnsq0wt8u)

[2.2.3 Supportability 9](#_jz91fvy7461k)

[2.2.4 Scalability 9](#_3tevipdcwvws)

[2.2.5 Security 9](#_2v1zcqolms81)

[2.2.6 Regulatory 10](#_z4piu8knmip0)

[**3. Feasibility Discussions 10**](#_x756aaobkhh8)

[3.1 Market & Competitive Analysis 10](#_hjx32o2tiab9)

[**4. References 11**](#_h6ksg8vj3362)

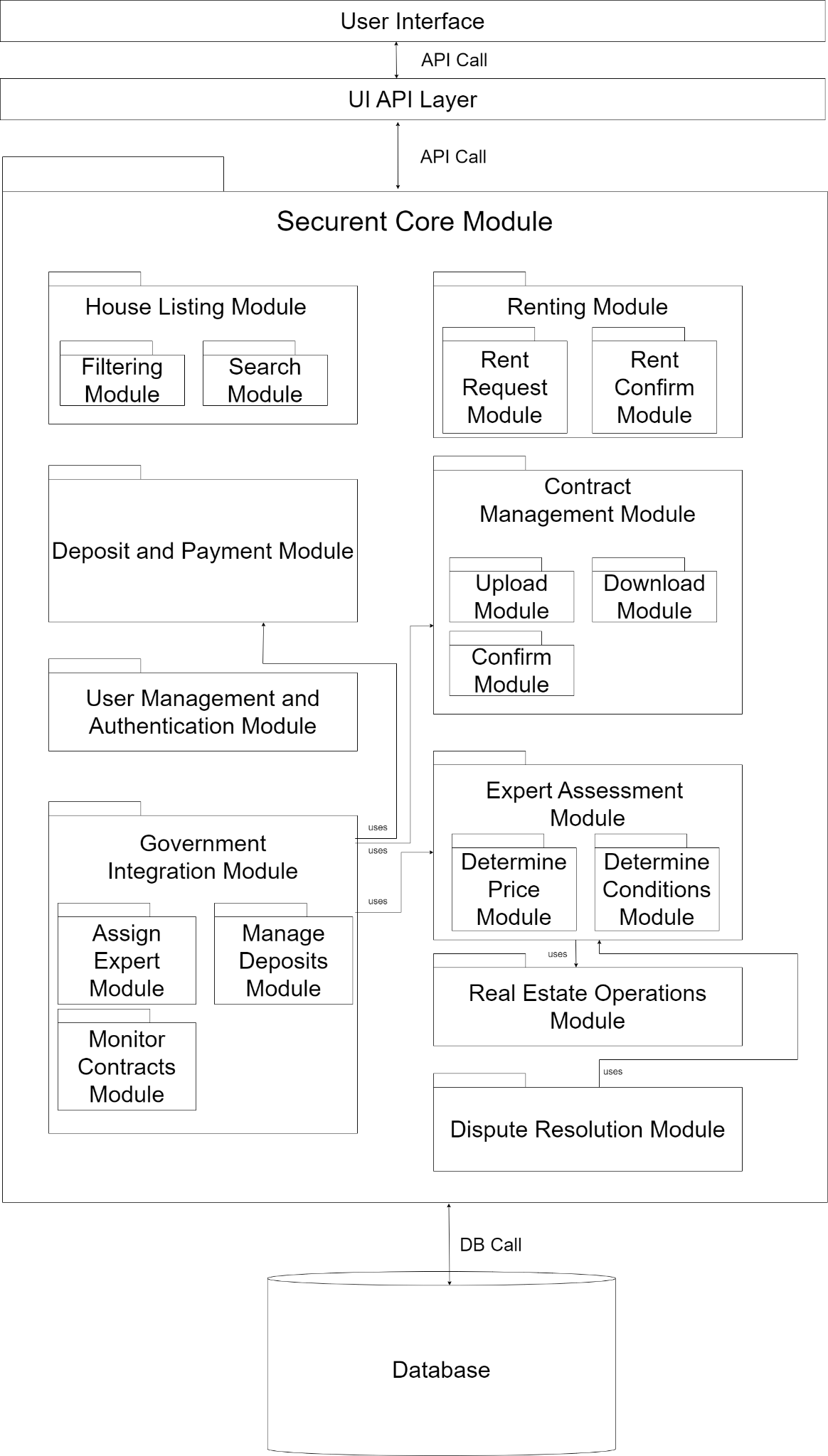
# **1. Introduction**

## **1.1 Description**

Our application SecuRent enhances the house rental process as well as addressing several issues that are related with the house rental process. SecuRent application emerges as a pioneering solution designed to transform the dynamics between tenants and landlords. With SecuRent, tenants can request officials (who are assigned by the government) to determine the rental price of a house, if they are not in favor of the price set by the landlord. Thus, ballooned prices set by the landlords can be regulated by the officials to prevent tenants paying more than what is worth for a particular house. Furthermore, if a house is rented, the status of the house will be updated in the application and a contract uploaded by the landlord will enable the government to monitor the contract. If a landlord wants to increase the rent price before the contract expiration date, the government will be notified with a request. Hence, the rent prices will not be increased without the government's consent, ultimately preventing unwanted skyrocketing prices that cause hardships for tenants. Moreover, with SecuRent, the deposit money for houses will be collected at the government treasury. The implication of this use case is that the deposit money will be given to either the tenant or the landlord depending on the condition of the house after the contract expires. Thus, conflicts between tenants and landlords caused by deciding who will keep the deposit money will be solved, as the government will act as the mediator in this process.

## **1.2 High Level System Architecture & Components of Proposed Solution**

The high level system architecture of the SecuRent is given in the next page. In the architecture, module style is used. Main features of the system are grouped into modules and their relationship with each other is displayed with arrows. Moreover, layered architectural style is used, since we will have different layers to respond to front end and back end requirements of the system, as well as API layers to maintain communication in between layers. Lastly we implement the use style, since particular modules use the functionality of other modules.



*Figure 1: High Level System Architecture*

### **1.2.1 Description of the Architecture Diagram**

* House Listing Module: Tenants and landlords can list and browse available houses for rent.
  + Filtering Module: Applies filters to the houses, such as district, floor, number of rooms, direction.
  + Search Module: Search functionality is covered within this module, as there will be a connection with the database to access the houses.
* Deposit and Payment Module: Deposits will be paid with this module.
* User Management and Authentication Module: In this module, authentication functionality will be covered.
* Government Integration Module: In this module, government related functionality will be covered.
  + Assign Expert Module: The government will assign experts to houses, with the functionality covered in this module
  + Manage Deposits Module: The government will collect and distribute deposits with this module.
  + Monitor Contracts Module: With this module, the government will monitor the contracts.
* Renting Module: The renting functionality will be covered within this module.
  + Rent Request Module: In this module, the tenant will request a house to be rented after negotiations are completed with the landlord.
  + Rent Confirm Module: The landlord will confirm the rent request of the tenant and the status of the house will change to the rented. Hence, the house will not be displayed to other tenants for rental purposes after the confirmation, since it has already been rented by a tenant.
* Contract Management Module: The functionality will be covered within this module.
  + Upload Module: Landlords can upload rental contracts to the system, and the government monitors these contracts.
  + Download Module: Upload rental contracts to the system can be downloaded with this module.
  + Confirm Module: The government will confirm the contract whether it belongs to the renting. If the uploaded document is an irrelevant document, the landlord will be notified to upload the contract.
* Expert Assessment Module: The expert judgments on the house to be rented will be carried out in this module.
  + Determine Price Module: The renting price of the house will be determined by the official assigned by the government. The functionality will be covered in this module.
  + Determine Conditions Module: The condition of the house will be determined by a real estate agent. The real estate agent will be chosen by the tenant to provide an objective judgment. Next, the conditions will be entered into the system by the real estate agent to help the government decide on which party will receive the deposit depending on the conditions.
* Real Estate Operations Module: Tenants can choose a real estate agent from multiple options in a specific area to assist in the rental process. Moreover, tenants can request a real estate agent to investigate the conditions of the house after the expiration of the contract.
* Dispute Resolution Module: Upon the real estate agent’s evaluation of the house conditions, the government will determine whether the deposit goes to the tenant if the house is in good condition or to the landlord if the house is damaged.
* User Interface: The GUI in which the clients will interact with the system.
* UI API Layer: The API layer which serves as the communication channel between the back end and the front end of the system. HTTPS requests are used to make calls in between layers.
* Database: The data related to the houses and contracts will be stored in the database. A relational database will be used and SQL queries are used to make calls to the database.

### **1.2.2 Stakeholders**

* Tenants: Individuals seeking to rent houses through the application.
* Landlords: Property owners looking to rent out their houses and upload rental contracts.
* Real Estate Agents: Local agents who assist tenants with the rental process and connect them with landlords. They can only serve in their reserved area. Moreover, they examine the house conditions after the contract ends.
* Government: Manages the pricing of the house rents, collects and monitors deposits, and oversees the rental process.

### **1.2.3 System Perspectives**

* Tenant Perspective: Tenants use the application to browse available houses, send rental requests, and make rental payments. They can also view the status of their rental deposits.
* Landlord Perspective: Landlords use the application to list their properties for rent, confirm rental requests, and upload rental contracts. They can also check the status of their rental deposits.
* Real Estate Perspective: The real estate agents use the application to check the requests they receive from customers to help them in the renting process and evaluate the house condition after contract expiration.
* Government Perspective: The government uses the application to manage the pricing assessments, collect and monitor deposits, and oversee the rental process for compliance and dispute resolution.

## **1.3 Constraints**

In this section, we will observe the implementation, economic, and ethical constraints.

### **1.3.1 Implementation Constraints**

* Government Collaboration: One of the stakeholders is the government. Hence, the government perspective will help the regulation of the house prices, deposit collection, and conflict resolution on deposits. However, the application assumes that the government will collaborate and use the application. Otherwise, the application’s aforementioned functionality will not be covered. Hence, we will contact the multiple ministries to advertise our application and their role in the application.
* Technical Infrastructure: The application aims to handle a large number of rental transactions, contract uploads, evaluation uploads. In addition, a huge amount of data will be stored in the database regarding the house information and contracts. Hence, there may be a need for large scale data storage solutions such as Amazon S3 in the large scale release of the application.
* Integration With Existing Systems: There may be a need for the integration of the application with the government databases, or payment systems, which introduce technical risks.
* Scalability: Given that the application will be adopted by potential tenants in Turkey, the system must be scalable. Moreover, the performance of the application must satisfy the users.

### **1.3.2 Economic Constraints**

* New Job Cost: In our system, the government should hire employees. The government should hire experts for identifying the rental price of the houses. On the other hand, the government may want to hire some people to track how the system works and the renting process goes. This will cause a new job opportunity under the government’s roof. Hiring new employees or allocating a workforce will bring costs to the government.
* Maintenance of the system: Since our system will cover the whole country’s rental system, it requires big databases and connections with other hardware and software systems. This will lead to an increase in the government’s expenses.
* Arrangement with Banks for Deposit: One of the main features of our project is to change where the deposits of the rentals go. To be able to make our system run through the country and the government can fund the costs of the system, deposits should be deposited into the government’s bank accounts. So as to provide this feature, the government needs to make arrangements with the banks. They should build a system in which people can put the deposit of the houses.

### **1.3.2 Ethical Constraints**

* Accurate and Reliable Costing: In our system, prices of the rental houses will be identified by the expertises that are assigned by the government. To be able to have people register into the system and not to victimize people, the prices should be set correctly.
* Trustworthy Assessment: When the tenant leaves the house, the deposit either will be returned to the tenant or handed to the landlord. The decision will be made according to the forms that will be filled by the real estate agents. While filling the forms they need to consider it by following ethical rules.
* Data Processing: To be able to create a system in which the whole rental processes run, we need to register all the rental addresses into the system. Since this kind of data is considered as private data, we need to handle it in an ethical manner. Thus, a new user who signed up to the system will be met with a terms of agreement to be able to use the system.

## **1.4 Professional and Ethical Issues**

* Government Oversight and Impartiality: Since we evolve the government into the system, it will work as a guarantor in the rental processes. The involvement can diminish the exploitations but the officials assigned by the government need to follow the ethical standards and impartiality.
* Data Privacy and Security: The address information, rental contract and financial transactions will be held through SecuRent. This kind of data is sensitive and needs to be handled wisely. On the other hand, the system contains information about the real estate agents. SecuRent should provide security and privacy to protect financial and personal data of tenants, real estate agents and tenants.
* Transparency in Decision-Making: While deciding the rental price of the houses by assessing various parameters such as the location, status, size of the houses; the decision making process should be transparent and logical. This is another ethical issue that SecuRent needs to solve. The problem can be solved by making both tenants and landlords understand the criteria of the assessment.

## **1.5 Standards**

### **1.5.1 Standards Used**

* UML: UML 2.5.1 is used through the draw.io platform to model the high level architecture of the system.
* Architectural Patterns: Module style is used to model the architecture. Specifically following styles are used:
  + Decomposition Style: Modules are separated into submodules to cover functionalities within the parent module.
  + Uses Style: Different modules connect to each other with the user relationship as particular modules require the functionality of other modules.
  + Layered Style: Different layers allow different functionalities to be separate. For instance UI and core module are distinct layers and interact with each other through an intermediary API layer.

### **1.5.2 Standards To Be Used**

* Software Development Standards: The software development of the SecuRent should be applied by following the reports and aiming for the actual final product. The standards of the development phase should be complied which involve the language and technology choices and predefined development processes.
* Usability Standards: The User Interface (UI) and User Experience (UX) of the SecuRent should be considered so that it follows the usability standards that are user friendly, intuitive, and easy to navigate in the system.
* Quality Assurance Standards: SecuRent should comply with the quality assurance standards that are satisfying the requirements of the project. Also the quality of the final product should be fine enough to be used as a rental regulatory application.

# **2. Design Requirements**

## **2.1 Functional Requirements**

* Users must sign up to the system by entering their credentials.
* Users must login to the system by using their email and password.
* State officials can create profiles and provide services.
* The tenant should be able to pay the deposit through the application.
* Tenants can pay the deposit amount through the application.
* Landlords can enter the rental price of their houses to the application.
* Landlords can enter the additional information of their houses such as property type, area of the house, room number, the floor where it is located, heating of the house.
* Landlords can upload images of their houses.
* The user can search the rental houses according to their region and address.
* The tenant can contact a landlord by using the contact information.
* State officials can restrict the rent increase rate monthly or yearly.
* State officials can detect the rent increases that do not comply with the regulation.
* Real estate agents can create profiles and provide services to tenants.
* Real estate agents can upload the form about if the deposit should be returned to tenant or given to landlord
* Real estate agents can view the rental houses in their location.
* Tenants can send a rent request to examine the house.
* Landlords can either accept or decline the rent request sent by the tenants.
* Landlords can upload the rent contract to the system.
* Government can confirm or decline the contract.
* Government can send officials to identify the price of the house.
* Government can return the deposit to the tenants.
* Government can give the deposit to the landlord if the house is damaged.

## **2.2 Non-Functional Requirements**

### **2.2.1 Usability**

To enhance the usability, the user interface must be simple and easy to understand. Users of the application can vary substantially. Users from all age groups (starting from 18 years old minimum) should be able to use the application without difficulty. Our aim is to make an application that most citizens of Turkey will use. Possible tenants should be able to access the particular home within at most 10 clicks.

### **2.2.2 Reliability**

As mentioned in the previous part, this application has the possibility of being used by tens of millions of people. For this reason, the application must be reliable and it should not crash in millions of user traffic.

### **2.2.3 Supportability**

SecuRent is aimed to be used for a long time. In its lifetime, new technologies may arise and some of the users might start to use these technologies. SecuRent should be accessible in these new technologies because it will have many users all around the country. Therefore, SecuRent may be needed to update frequently. So, SecuRent should be implemented in a way that makes later updates easier.

### **2.2.4 Scalability**

SecuRent is aimed to be used by all the tenants, landlords and real estate agents in the country. Therefore, it will already be used by many users beginning from the first day it becomes live. Also the number of the potential users might increase as the population of the country increases or more houses are built in the long term. Hence, SecuRent should be scalable to be able to increase the capacity of the servers when it is needed.

### **2.2.5 Security**

SecuRent collects countless sensitive data such as Turkish Republic identity number, address and phone number of the users, contract information, etc. All of this data must be protected by encryption. Tenants will also perform payments through the application. Money transactions must also be secure so that no user loses their money.

### **2.2.6 Regulatory**

Some of the functionalities of SecuRent such as collecting deposits through a third party system might violate current regulations by the government. As the customer of this application is also the government, current regulations can be changed.

# **3. Feasibility Discussions**

## **3.1 Market & Competitive Analysis**

In the market, there is no robust competitor for this project. There are similar projects that include some features of SecuRent, but these projects do not function similarly and their aims are not the same. On the internet, we could find 2 projects/websites that are in the same application domain as SecuRent: Endeksa and mydeposits.

Endeksa is a service which offers the users to rent, buy, or rent a house. It is supported with machine learning which predicts the market price of a house. The website has some fields to fill in when a landlord creates a publication of a house to be sold. These fields include specific information about the floor, facade and the condition of the flat. Users can also examine in a specific location to look for the renting/selling opportunities [1]. Though it is in the same application domain as SecuRent, the contract system, interactions with the government and the depositing system is different from SecuRent. So, in practice Endeksa is not a challenging competitor for SecuRent since they function differently and they offer different services.

Mydeposits is a service which is constructed by private rental sector experts. It is a deposit protection service which landlords, agents and tenants can use. The website is dedicated and designed to protect the deposits mainly, so there is no governmental profit from these deposits or detailed tracking of contracts [2]. Regarding these missing features, mydeposits is not a powerful competitor for SecuRent since the functionalities and scope is different.

Note that for this competitive analysis these 2 websites could be found and assessed, and there may be different websites that may be a potential competitor. After thorough internet research regarding Turkey, it can be concluded that SecuRent does not have a competitor, since SecuRent is not a product but a new service to be offered.

# **4. References**

1. *endeksa.com*. [Online]. Available: https://www.endeksa.com/tr/. [Accessed: Nov. 15, 2023].
2. *mydeposits.co.uk*. [Online]. Available: https://www.mydeposits.co.uk/join-now/. [Accessed: Nov. 16, 2023].