METU Community Services

System Design Document

v1.1

29.12.2013

Rebellion

Barış Güvercin 1746031

Burak Çelik 1746536

Eren Deniz Çelebi 1746528

Taylan Doğan 1819259

Table of Contents

1. Overview

1.1 Scope

The document holds the structural overview of all modules, interfaces, data and module designs in order to support design and development process. In the implementation of the process, this document will be a direction for developers.

1.2 Purpose

This document is prepared to describe and visualize the basic architecture of METU Community Services application. The main aim of this document is to identify the software system which is designed to meet the requirements of the Software Requirements Specification document.

1.3 Intended audience

The expected audience for this document is the development team of the software. The team can use this document for reviewing and implementing purposes.

1.4 References

- StarUML 5.0 User Guide. (2005). Retrieved from http://staruml.sourceforge.net/docs/user-guide(en)/toc.html

- IEEE. IEEE Std 1016-2009 IEEE Standard for Information Technology – System Design – Software Design Descriptions. IEEE Computer Society, 2009.

2. Definitions

The definitions of the terms, used in this document, are shown below:

Terms	Definitions
E-R Diagram	Entity relationship diagram
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
SDD	Software Design Document
UML	Unified Modelling Language
SRS	Software Requirement Specification

3. Conceptual model for software design descriptions

3.1 Software design in context

This project will be designed as object-oriented paradigm with using Android SDK via Eclipse. In this structure, the application will be fast for not using Android tools that will be in front of SDK and the user. Moreover, the new ideas are being developed each day by our team and making the application this way makes it easy to response to the needs of the user.

This document consists of viewpoints of the application such as context, composition, logical, dependency and interface. The reason for this to be prepared is to give developers ideas about what would be the application architecture and what to expect from them. In other words, this paper is prepared to understand system designs that is lacking.

Each design viewpoint determines the type of elements that will be used by developers throughout the project. Every viewpoint has its own elements and expression, thus all of them will be independent in terms of guidance of the aspect.

3.2 Software design descriptions within the life cycle

3.2.1 Influences on SDD preparation

The key software life cycle product that drives a software design is typically the software requirements specification (SRS). The requirements in the SRS and the demands from the users who will use the application

3.2.2 Influences on software life cycle products

In this project, services like food, calendar, weather and community should be reachable from the menu as soon as the user starts the application. The order of the development of these services is not important. However, when showing the beta demo to some people, all of them must be functional at some point of the system. The development will continues as from this point by the needs of user for positive feed-back.

3.2.3 Design verification and design role in validation

Verification and validation will be tested after the preparation of the test cases. All system parts will be tested against these cases. In this process, it will be checked whether the requirements in SRS are fulfilled or not.

4. Design description information content

4.1 Introduction

Software Design Description of METU Community Services introduces how this application will be designed and implemented. During the design of this project's modular object-oriented structure, practical and qualified interface will be designed.

4.2 SDD identification

This application is being built by Team Rebellion. It will be used for reaching many services inside the METU campus like transportation, venues, etc. Also, METU Community Services will be a great example for other universities. Thus, it will help them to develop similar applications.

4.3 Design stakeholders and their concerns

METU Community Services' stakeholders are the people who live, work or study in the METU Campus. Stakeholders' possible concern is validity of the information we provide through the services. In other words, the reachable information through this application must be updated so that it is true at all times.

4.4 Design views

As the object-oriented design is chosen for this project, the project can be updated easily. Once the application is downloaded and installed through the online market, it can be started by simply touching the application icon. After starting the application, the users will find themselves in the main menu, in which the service icons are listed.

The user can navigate through the different services by touching these icons. Once an icon is touched, the corresponding service menu will be shown to the user. By pressing/touching the “Back” button of the device, the user can get back to the main menu easily. And if the user wishes to exit the application, it will be enough to press/touch again the device’s “Back” button, while in the main menu. A logical view of the application is explained. State dynamic view shows the state transitions with a diagram. The interface view is really clear and supported by figures.

4.5 Design viewpoints

Viewpoints show what is expected from the user and give a detailed description of the whole system. The information will flow between the user and the system. Relation of inputs and outputs will be explained in the context viewpoint. Logical viewpoint shows the class structure. The basic structure of the designed application will be presented in the composition viewpoint. Details of internal and external interfaces will be provided in the interface viewpoint. And in the state dynamics viewpoint, the behaviour of the system will be explained.

4.6 Design rationale

Design choices are made, taking some significant features like sustainability into account. According to the users’ and stakeholders’ requirements, it can be updated. Also, each function in the software will be commented, since it is important to do this for other developers, in terms of readability.

4.7 Design languages

Unified Modelling Language (UML) is selected as a part of design viewpoint specification.

5. Design Viewpoints

This section will give brief knowledge about design viewpoints.

5.1 Introduction

Several design viewpoints in terms of design concerns for use will be defined in following subsections. UML shall be used as a design language.

5.2 Context viewpoint

The context viewpoint is used for describing relationships, interactions and dependencies between the user and the system. The use case diagram is mostly responsible for showing relevant information between the actors and the services.

5.2.1 Design concerns

The purpose of the context viewpoint is to be crystal clear in the field of services, operations and design scopes concerning the project. This part is obviously a key to development since it mostly investigates the relationship between actors and the services that is offered by the application, thus making it applicable to most design efforts.

5.2.2 Design elements

The major components of design elements are design entities, design relationships and design constraints. For example, the actor of user, in use case below, can see relevant information while other actors support the services that prepare this information. They have this kind of relationship between them and the environment, like receive and provide information.

5.2.3 Example languages

The relevant UML use case and context diagram for this section is as follows:

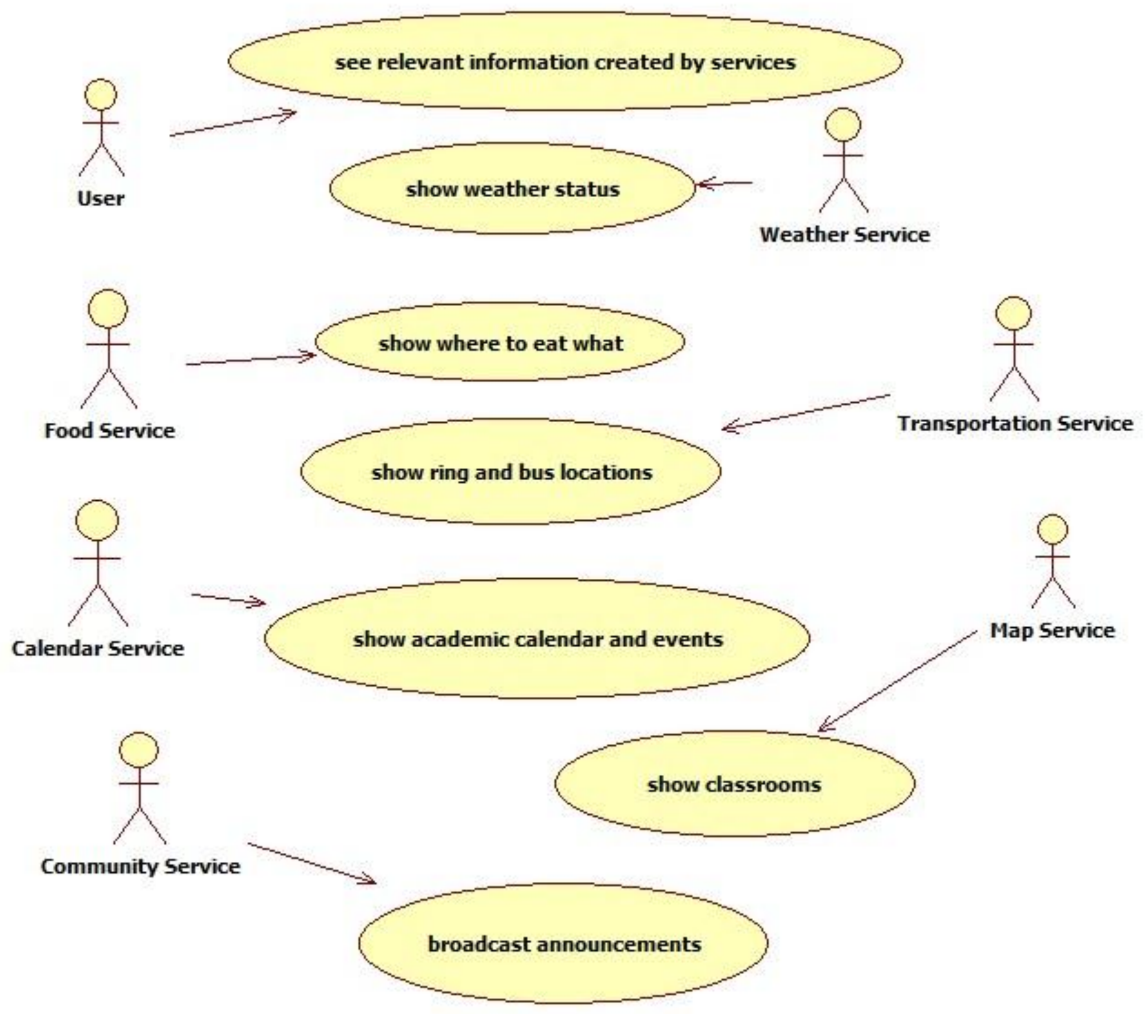


Fig.1. Use Case Diagram

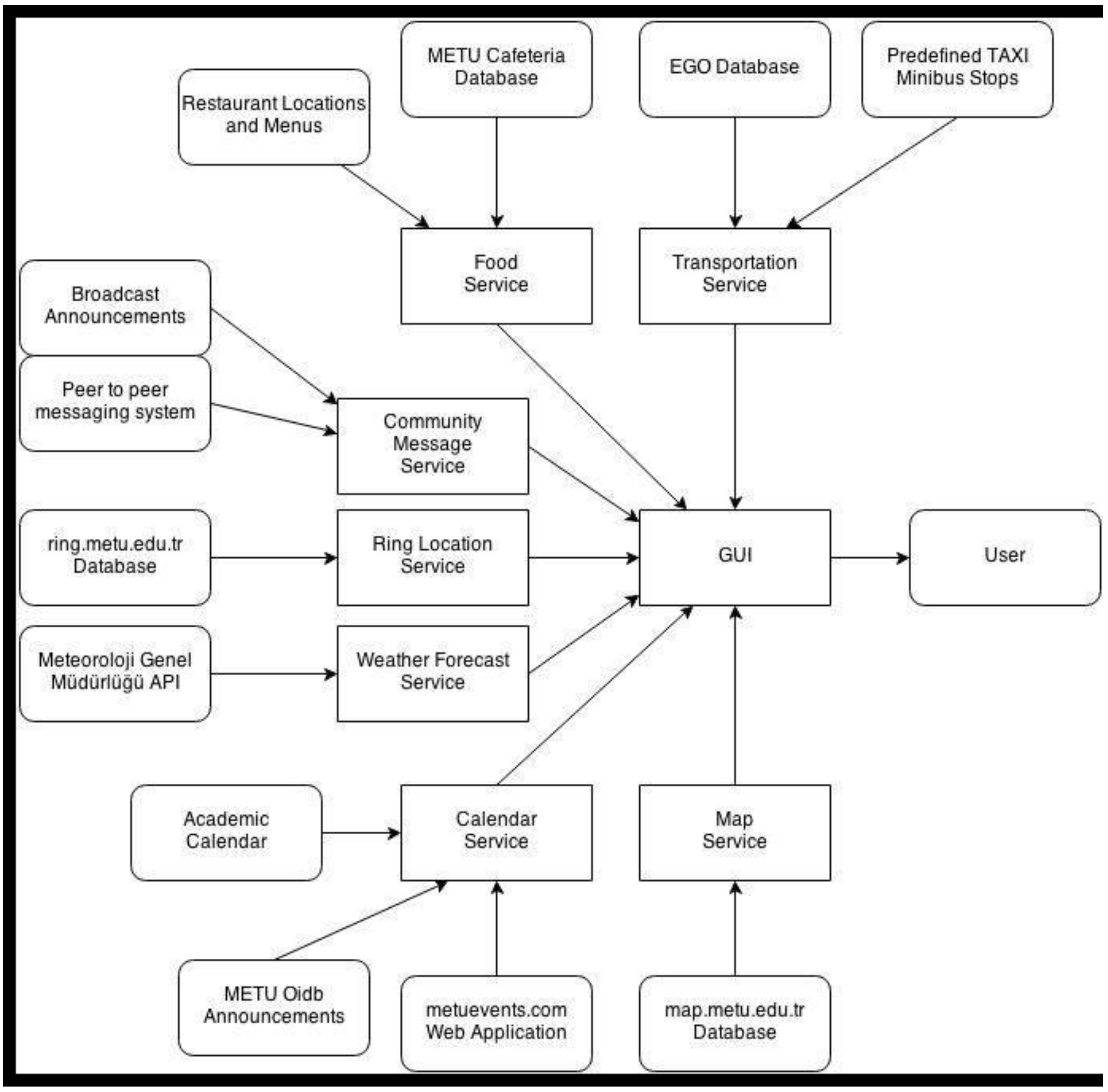


Fig.2. Context Diagram

5.3 Composition viewpoint

This part will show the basic structure of the system to be designed.

5.3.1 Design concerns

Composition viewpoint of the METU Community Services includes components of the application and determines the relationships of that system components.

5.3.2 Design elements

As design entities, there are seven services are taken place. A user can take the information they want from relevant service. Components will be shown in Component Diagram below.

5.3.2.1 Function attribute

Each service that the application provides gets information from distinct databases. These databases are used for public information services. User interface of the application will navigate the user to the information that user wishes to get.

5.3.3 Example languages

The relevant UML component diagram for this section is as follows:

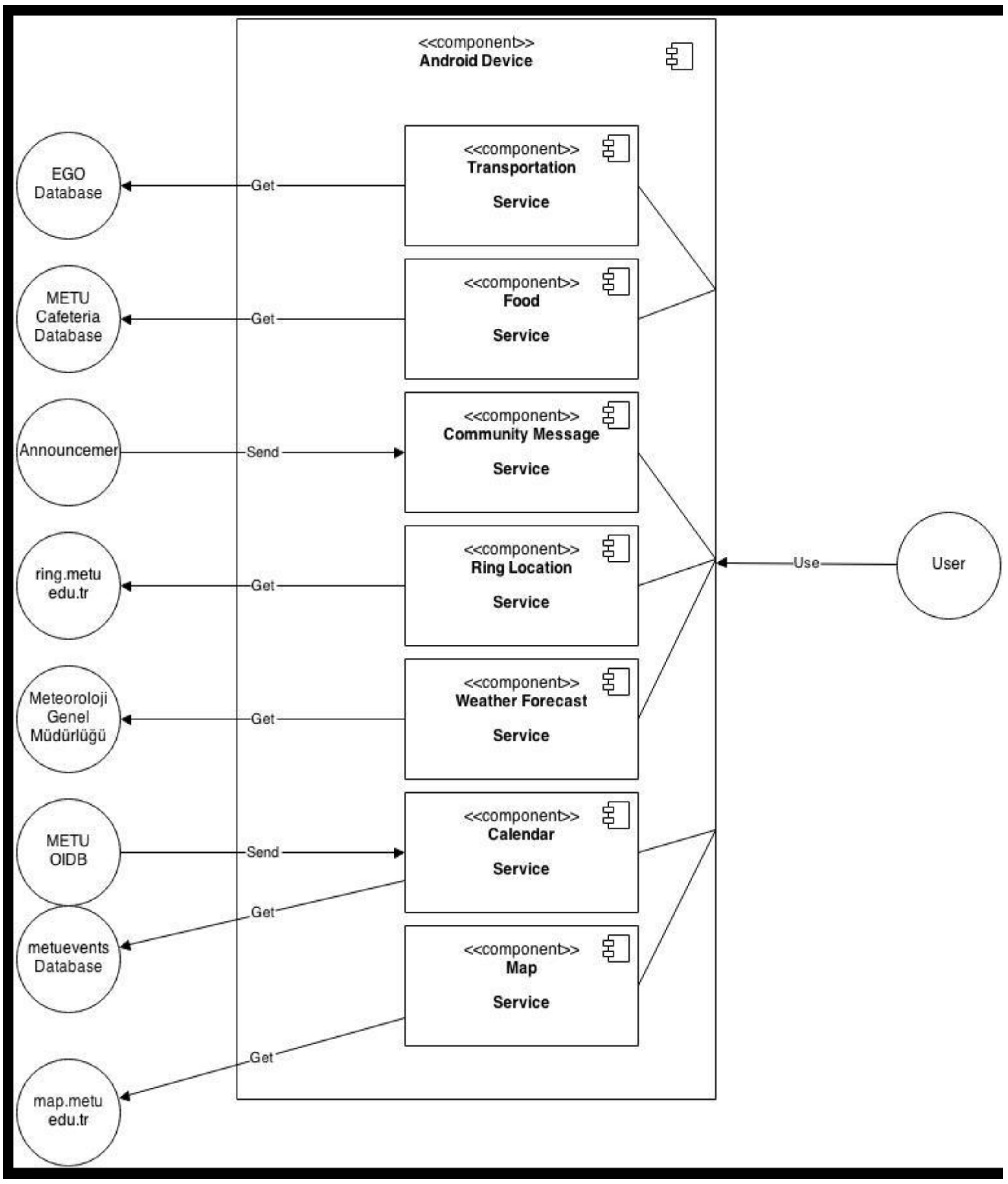


Fig.3. Component Diagram

5.4 Logical viewpoint

The logical viewpoint of the system aims to describe the implementation as classes with their structural relationships.

5.4.1 Design concerns

Logical viewpoint shows the system by analysing the classes which are constructed at the implementation step of the application.

5.4.2 Design elements

Design entities, design relationships, design attributes and design constraints are stated in the following section. Briefly:

Transportation Service: This class will hold taxi and minibus addresses.

Food Service: METU Cafeteria will be handled as "cafeteria" and other restaurants and cafes will be represented as "restaurant". Restaurant data is stored in the "Restaurant" class.

Community Message Service: Broadcast announcements will be handled in this class. Only privileged users, such as METU BIDB, can do broadcast announcements (urgent messages such as water cut). If a user registers to system s/he can use Peer-to-Peer messaging service and join a group, otherwise s/he will get the broadcast messages. After the core components of this class are developed, we can add new features like friending, photo sharing etc.

Ring Location: This class will get the whole data from ring.metu.edu.tr.

Weather Forecast Service: Meteoroloji Genel Müdürlüğü API will be used for provide this service. This will show three days' weather data (temperature and weather condition).

Calendar Service: METU's official events, like announcements from OIDB or student societies will be delivered to the users. Users can get all information about an event. In addition to this we will get the data from metuevents.com for the unofficial events that concern METU people.

Map Service: The data will be pulled from map.metu.edu.tr. This aims to show users (mostly freshmen, people came to an exam) proper places of the classes and departments.

NOTE: connect() function handles the connection between our application and relevant database. This may differ service to service; it may be constant connection or time-delayed connection.

5.4.3 Example languages

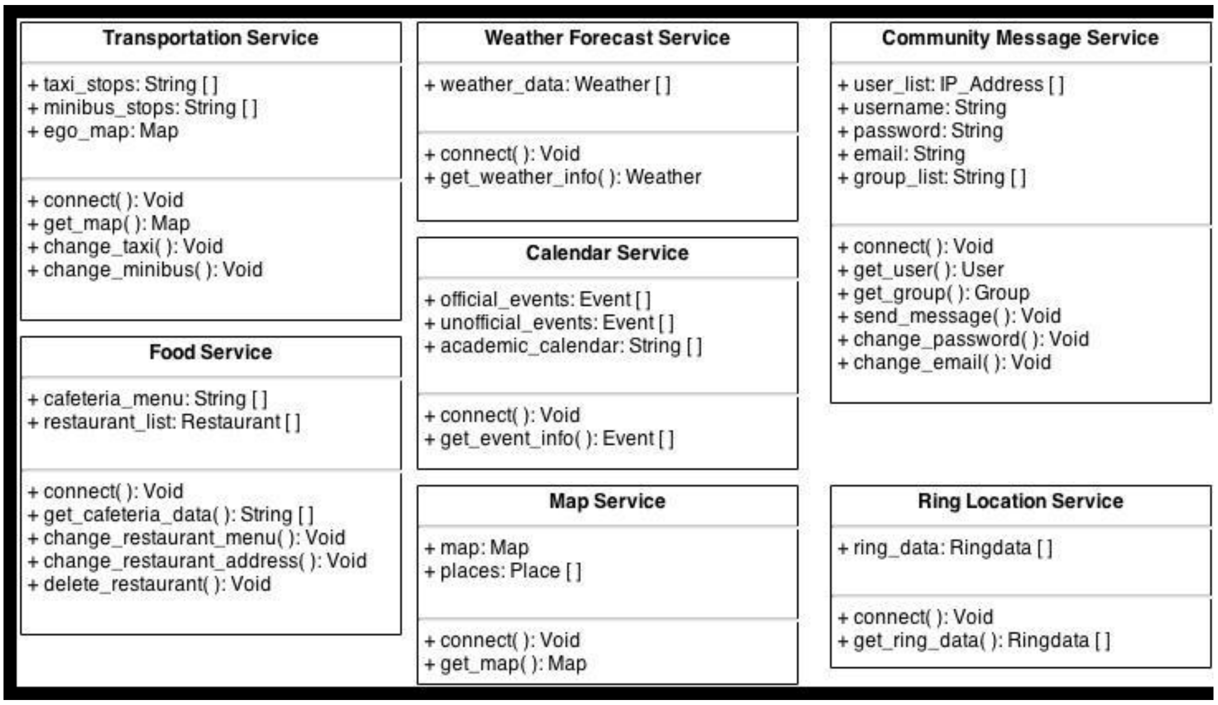

Transportation Service
+ taxi_stops: String [ ]
+ minibus_stops: String [ ]
+ ego_map: Map
+ connect( ): Void
+ get_map( ): Map
+ change_taxi( ): Void
+ change_minibus( ): Void
Food Service
+ cafeteria_menu: String [ ]
+ restaurant_list: Restaurant [ ]
+ connect( ): Void
+ get_cafeteria_data( ): String [ ]
+ change_restaurant_menu( ): Void
+ change_restaurant_address( ): Void
+ delete_restaurant( ): Void
Weather Forecast Service
+ weather_data: Weather [ ]
+ connect( ): Void
+ get_weather_info( ): Weather
Calendar Service
+ official_events: Event [ ]
+ unofficial_events: Event [ ]
+ academic_calendar: String [ ]
+ connect( ): Void
+ get_event_info( ): Event [ ]
Map Service
+ map: Map
+ places: Place [ ]
+ connect( ): Void
+ get_map( ): Map
Community Message Service
+ user_list: IP_Address [ ]
+ username: String
+ password: String
+ email: String
+ group_list: String [ ]
+ connect( ): Void
+ get_user( ): User
+ get_group( ): Group
+ send_message( ): Void
+ change_password( ): Void
+ change_email( ): Void
Ring Location Service
+ ring_data: Ringdata [ ]
+ connect( ): Void
+ get_ring_data( ): Ringdata [ ]

Fig.4. Class Diagram

5.5 Interface viewpoint

This description includes the details of external and internal interfaces. This viewpoint provides information to programmers and designers for creating interface from a more correct perspective.

5.5.1 Design concerns

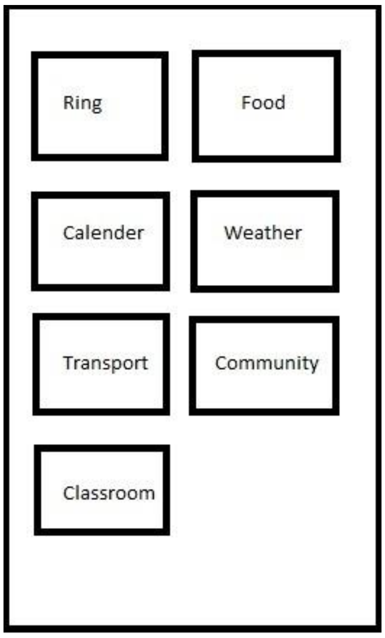
The interface view description is important in this project because this is a mobile application and everything depends on GUI. A clear description of the entity interface is essential for smooth integration of the project development.

5.5.2 Design elements

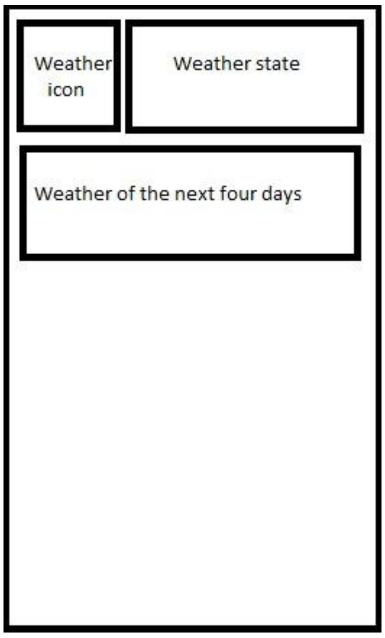
The application consists of several windows. The starter one of them is the main screen. The user can see the services that are provided by the application. He/she can reach ring, transport and classroom information by clicking the related buttons and receive a map for them. Moreover, the weather, food and the calendar service can be reached the same way in order to get information about those services. For each venue in food screen, you can get a different window with menu and prices for that particular item. Also, in the calendar screen the user can get related items for academic calendar and events.

5.5.3 Example languages

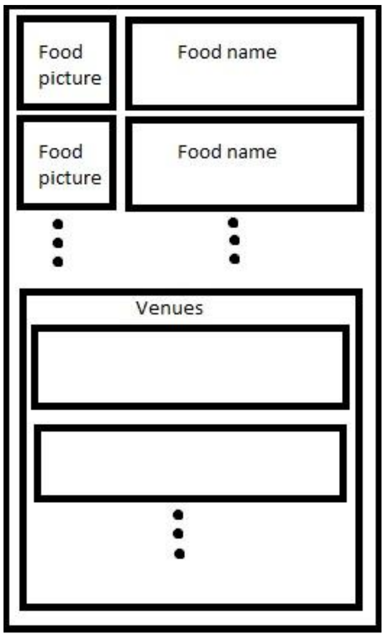
The interface system of our application which is explained in 5.8.2 is as follows:



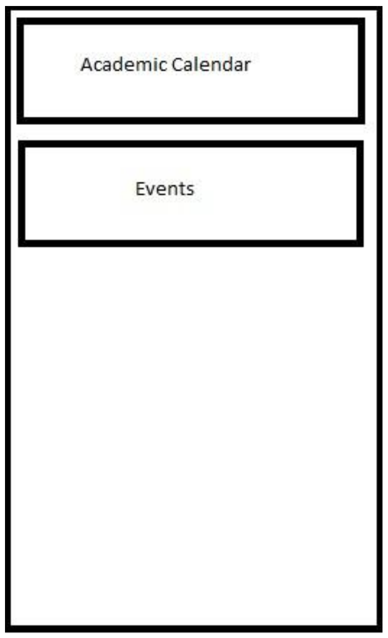
Main Screen

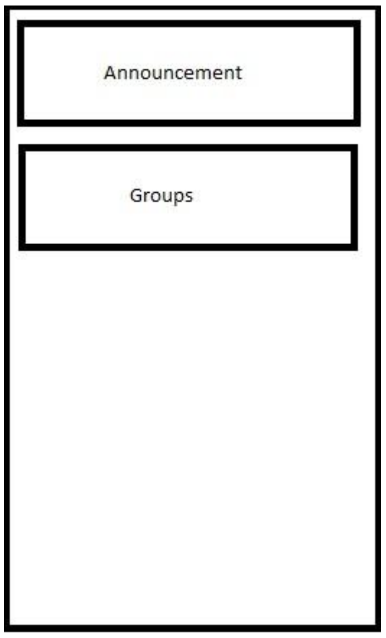


Weather Screen



Food Screen





Calendar Screen Community Screen

5.6 State dynamics viewpoint

State viewpoint deals with behaviour of the system when some particular action happened in the program flow. It shows how the application reacts to that action.

5.6.1 Design concerns

This viewpoint is basically about states and reaction of those states to events.

5.6.2 Design elements

The viewpoint consists of states which gives information about program status before user interacts with the application for more action and the transition between some particular actions. Moreover, the state flow and the critical region are described in the diagram for further implementation.

5.6.3 Example languages

UML state diagram of the project is below:

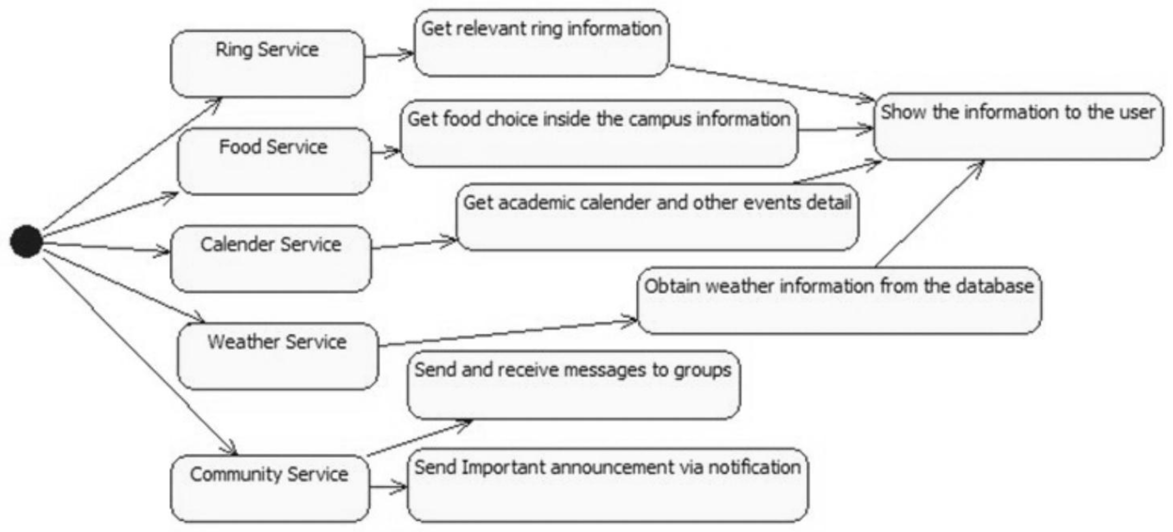

Ring Service
Get relevant ring information
Food Service
Get food choice inside the campus information
Show the information to the user
Calender Service
Get academic calender and other events detail
Obtain weather information from the database
Weather Service
Send and receive messages to groups
Community Service
Send Important announcement via notification

Fig.5. State Diagram

6. Planning

We intend to complete raw versions of components like weather, calendar, food and map service at the end of first semester. The users will be able to see things like the weather forecast, food choice of the cafeteria and events happening inside the campus etc. The distribution of this work among our team members is as follows:

- Burak Çelik – Weather Component

- Barış Güvercin – Calendar Component

- Deniz Eren Çelebi – Food Component

- Taylan Doğan – Map Component

At the beginning of the second semester, we will continue to implement unfinished parts that are left from the first semester. After we finish those components, we will go over the community component which is basically social part of our project. However, right now we are stuck at the ring development part because we could not be able to arrange a meeting for getting database sections of rings. As of from the second semester, we will try to deal with this situation more intensely. Furthermore, at first our transportation service contained government buses information but right now it is cancelled and our efforts or contacts with the government about this thread reached a dead end. Likewise, we will try to solve this issue at the begging of the second semester.

7. Conclusion

This document provides implementation details of METU Community Services up to a point. The basics of data design, modules and design viewpoints of the system are described. The main purpose of this paper is to acknowledge the developers of the important parts that they will be dealing while developing the project.