Bilkent University

Department of Computer Engineering

Object-Oriented Software Engineering Project

CS 319 Project: Tempo

Design Report

Project Group: 1.G

Member Names: Mert Saraç, A. A. M. Jubaeid Hasan Chowdhury, Burak Erkılıç, Kaan Kıranbay

Course Instructor: Eray Tüzün

Progress Report
March 03, 2018

Table of Contents 1

Analysis Report

CS 319 Project : Tempo

1. Introduction

In the first section of the design report, we will start with the explaining our purpose of system. Then we will mention our design goals in detail. We have explained some keywords, acronyms, and abbreviations about the system too. These are the main things that are considered while making the rest of the report.

1.1 Purpose of the System

Tempo is an application that aims to ease the life of the users by helping people to choose time for any kind of events for themselves or between their friends or coworkers, and by reminding people that an event that they should do or they want to do is approaching . The user interface is designed to maximize users' desire to use the application by its' crisp and smooth visuals and its' easy usage. The application will use internet to get information about events and users' friends. These feature makes the application kind of a different from other calendar applications which makes this application more enjoyable to use. Therefore, the purpose of the system is to deliver users a different experience for arranging events.

1.2 Design Goals

Main goals of this application's designs are simple: ease of use and satisfactory visuals. To accomplish these we have to trade some other important design components. In this part, we will talk about what our design goals are, what they mean to us, and the trade-offs.

End User Criteria:

Ease of Use: It can be advocated that people do not want to waste too much time to organize their daily routine so, easiness of the system is one of the core features. To makes organizing process faster, Tempo provides some mouse actions such as moving events to another time, creating new event by dragging specific time period in the calendar. In addition, having one main screen that contains many useful lists like To-Do-List and Friends List keeps all of the core functionality in the home page and allow quick access.

Ease of Learning: Since our program is designed for diverse group of age, ease of learning is important. In this way, our design is similar with the popular programs that users are used to due. Even though, in case of they are not used to some features, there is a help button to explain all features of the program such as creating events, explanations of events etc.

Satisfactory Visuals: To attract and encourage users, the program has many graphical features. All events have a color and it is changeable and various stickers can be attached to describe a event in a better way. Also, users can set a background wallpaper for calendar. Finally, night mode is available for users who wants to use the program at night so, brightness of the program does not hurt their eyes by changing default color scheme from light to dark.

Increased Productivity: As increasing productivity is one of our main goal, a lot of the designs are based on that. One of the core feature is smart event in which the user can invite all of his friends for group events such as birthday party without individually contacting them. This allows for arrangement of events for large groups without having to waste any time. Also notifications before any event aids the user to prepare for the event in early and attend a

event which he otherwise might forget. Also to-do-list allows the user to group subtasks such as shopping list that can aid the user to make sure he has completed a main task.

Maintenance Criteria:

Reliability: System should be purified from serious bugs and system crash problems. In order to achieve this, many testings will be done and while the implementation of program, all cases are approached carefully. In addition, using Singleton prevents some potential bugs in the system. For example, it is not possible to create more than one instance of Network class. However, if the system will crash, system send a system crash report to inform developers. Another problem is disconnection from internet and by having an offline mode, this problem is prevented partially. User cannot finalize a SmartEvent until becoming online. (Singleton)

Modifiability: To maintain a system, adding new features and updates are important to not lose users' interest and object oriented programming enables to easily change and add features without any bug. In addition, our program is open system for adding new features like chat between users.

Adaptability: In order to use the system in various platforms, the system is implemented by Java which is provides an easy way to implement for cross-platforms. The system will work without any problem in all JRE installed platforms. It is the main reason that we preferred Java in our implementation.

Performance Criteria:

Efficiency: Tempo's system is a responsive system which means it should be fast when users want to use or change anything in the system. Therefore, the program won't use much memory. The system's memory usage will be less than 10 megabytes. The system is able to do this by not using big pictures such as high quality pictures. The responsive system is helpful while using the database too because the system will be getting information from the users, sending that information to database and saving it, and if needed the information will be

received from the database. These informations will not have need large memory spaces to be stored. Therefore, the responsive system and efficient system will make this program's internet usage as low as possible like 1 Kbps download and upload.

Response Time: The system of Tempo uses little memory and internet to run the program. This helps program to run fast and have small latency. Some core features like adding a personal event or changing the time of the events or registration to the system will be done in at worst 100 milliseconds. However, some things like adding a friend or adding a Smart Event and inviting people to it will be kind of a slow compared to other features like adding a personal event to the calendar; because when adding people or adding smart events, the system has to send information to database and from database the system will use that information to send invites to the invited people. However, these won't be that slow like 2 seconds but it will take the twice or the 3 times time.

Connection Time: As already mentioned on the previous part, the system uses little memory and internet. This helps the system to connect to database and gather information with low latency. However, there is 2 more like connecting for the first time and connecting after a disconnect.These two will be treated together as just connecting to the system. Connection time to the system will be at worst 500 milliseconds.

Trade-Offs:

Usability vs. Functionality: As a team, we concluded that we should have a basic system for the sake of users because the users will be everyone. Therefore, the system should not be complex to use. To accomplish that we have to use not complex and not much functions. This will help users to learn and remember the functionalities of the system easily. All of these will make users enjoy using the system.

Performance vs. Memory: The system is planned to be efficient and responsive which means high performance and low latency. To accomplish those we have to sacrifice memory. Even though we are not using that much memory, it can be said that we sacrificed memory for the system's performance.

Efficiency vs. Reusability: In the system, one of the main goal is efficiency since reusability is not one of the concerns of us. In our design, none of our classes will be used for another project so, our design will be more functional by having simple implementation.

1.3 Definitions, Acronyms, and Abbreviations

Since the project has many features, there are some terms that are should be explained:

Smart Scheduling: This feature makes people be able to use their time more efficiently with their friends or co-workers by showing the common time among people to the users.

Event: This feature is the core feature of Tempo. An event is simply an activity that can be whether with your friends (that is called SmartEvent) or without them (that is called Personal Event). In addition, it has some other categories like timeless event, timing event, temporary event and permanent event so, Tempo enables users to combine these categories for describing their event in the best way.

Smart Event: This feature is an event that is created by one of the users' requests to other users who are attending this meeting. For this, when one user is scheduling, he/she can see who is available during this period so he/she can easily change the time period for other individuals. When decided on the time, the user request for Smart Event and then, other participants are notified for this meeting, its explanation, participants, time period. Other participants can whether accept this request or reject. When one participant accepts it, the system waits for other participants response. If all of them accept it, this Smart Event is added to all participants schedule. If any participant does not accept it, system inform the one who creates the event for who did not accept and asks "Do you want to still arrange this event?" without the one who did not accept.

Personal Event: This feature is an event that user's friend list cannot see what user is doing on this time period but they can see he/she is not available during this time period.

Timeless Event: This event type is that is can happen anytime in a day. In simple words, there is no specific time period to do this event like drinking water.

Timing Event: This event type has specific time allocated time for doing this event. For schedule this, user use the weekly-time table.

Temporary Event: This event type is one-time event unless user specifies it will repeat.

Permanent Event: This event type is repeated so when user adds this event to his / her schedule, it will remain until user decides to this event will not be repeated anymore. For this

feature, user can adjust how often this event will repeat like every day, every other day, weekly etc.

Abbreviations:

- **JRE** : Java Runtime Environment. It's a software package that contains what is required to run a Java program.[1] Therefore, JRE needs t o be installed on your computer in order to run Java applications.[2]

1.4 References

[1] https://en.wikipedia.org/wiki/Java_virtual_machine

[2] https://techterms.com/definition/jre

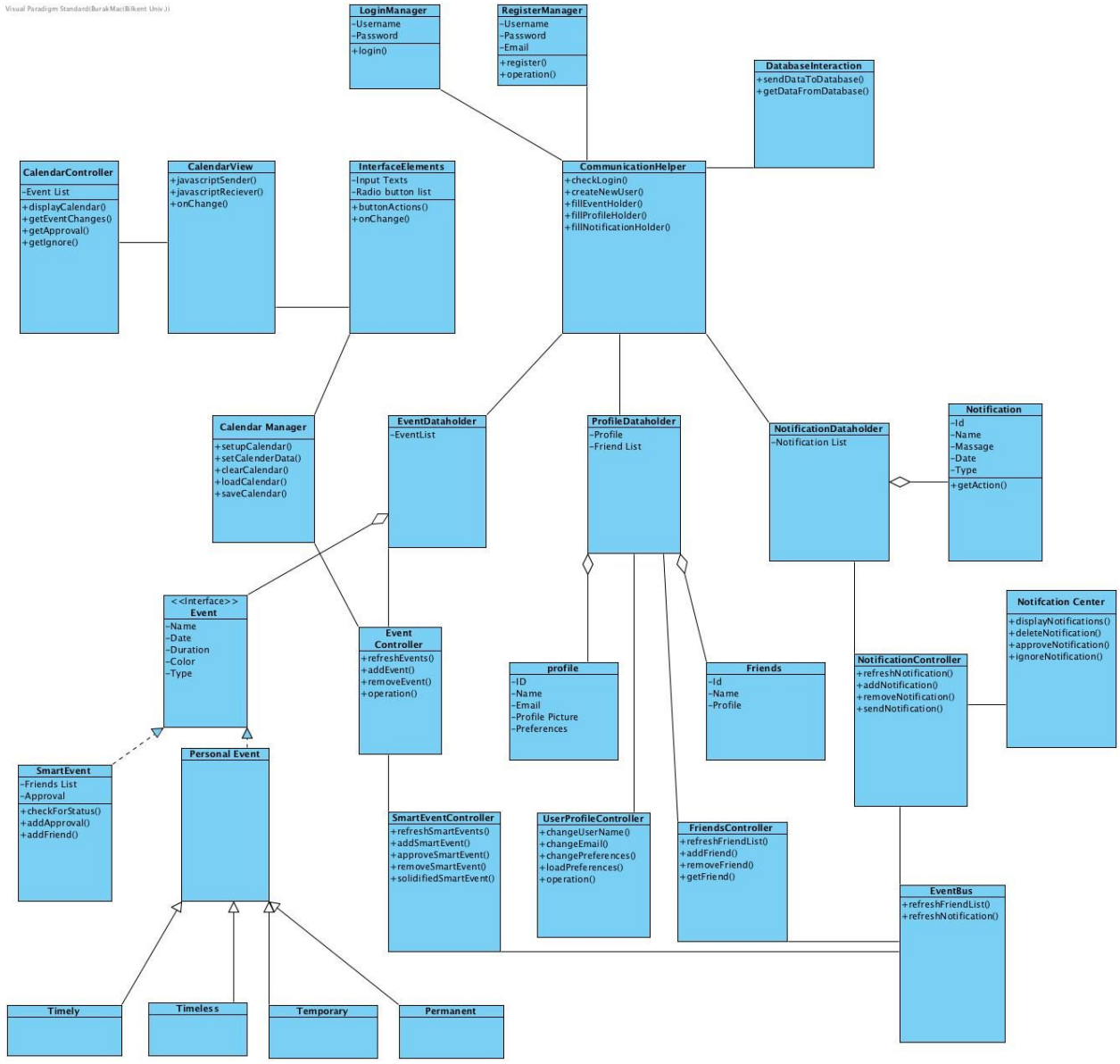

Figure: Detailed object design

2. Software Architecture

In this section, the decomposition, hard/software mapping, persistent data management, access control and security, and boundary conditions will be explained in detail. Main goal is

here to reduce the coupling between the different subsystems of the system, while increasing the cohesion of the subsystem components.

2.1 Subsystem Decomposition

Our design contains 6 main components that are designed for various features of the system. These are Authorization, Data Management, Event management, Profile Management, Dynamic Notification Management and SmartEvent Management due to it contains many features of other components as it is shown in Figure 1.

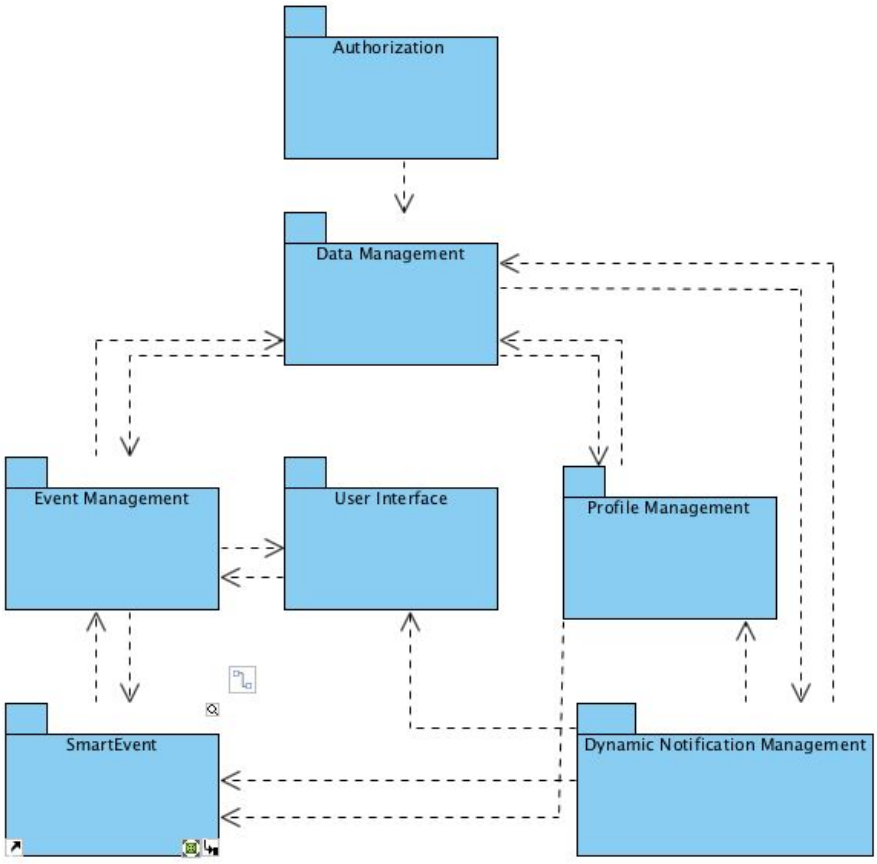

Figure 1 - Representation of Subsystem Decomposition

In Figure 2, it can be shown the overview of the all layers.

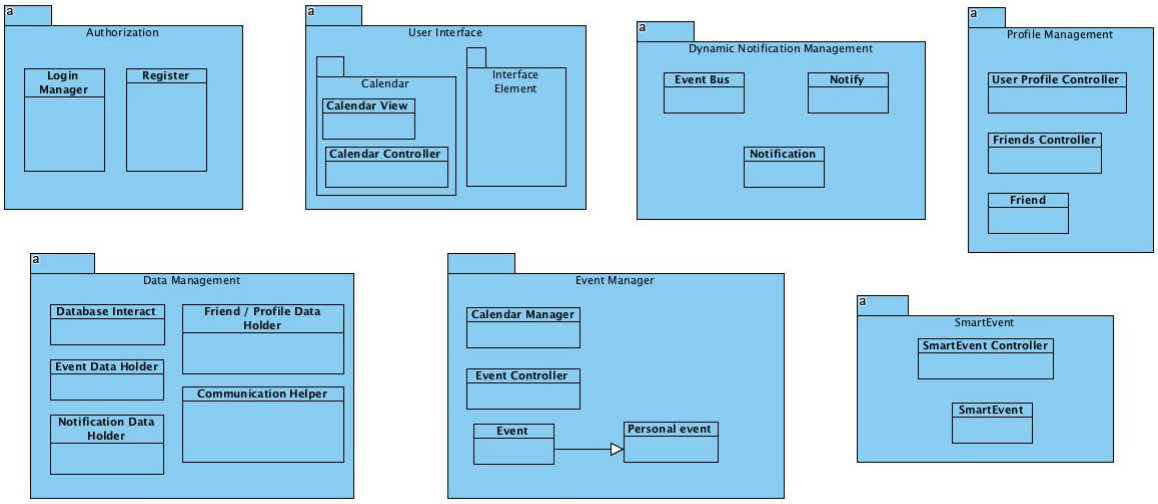

Figure 2 - Overview of Layers

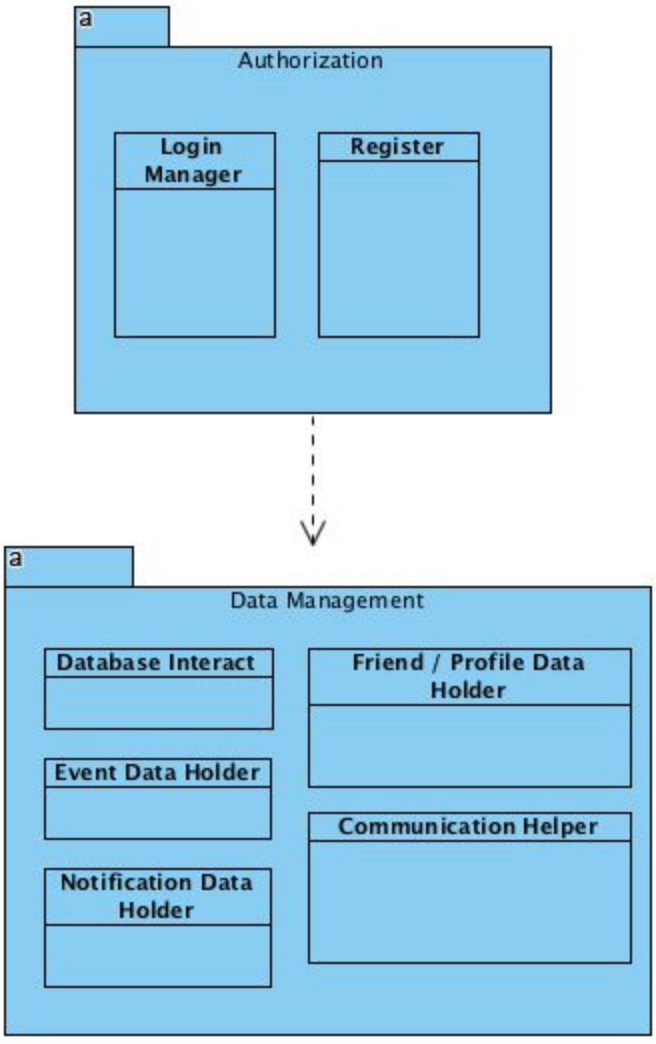

Figure 3 - Layer 1

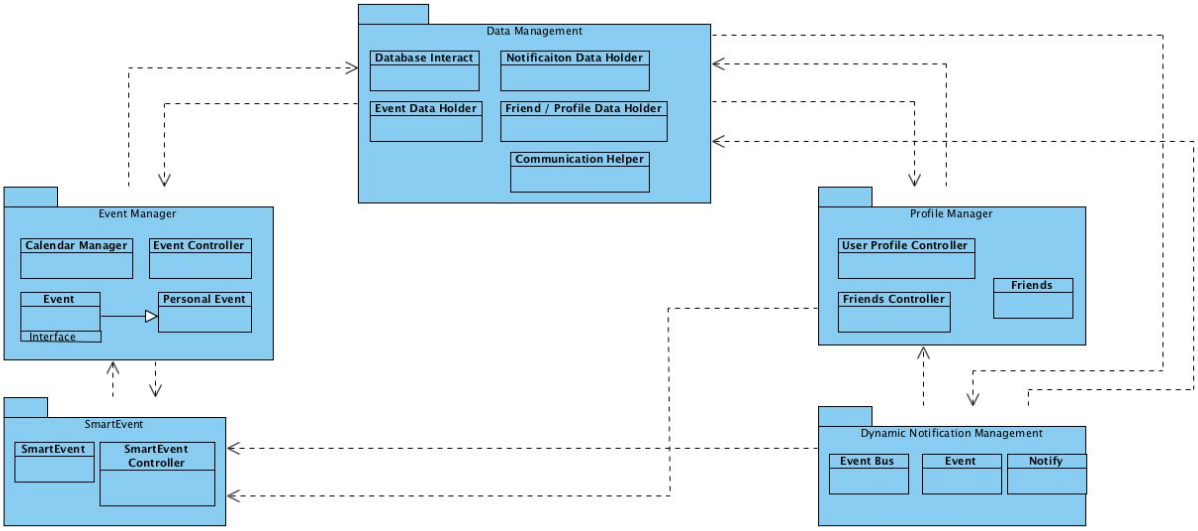

Figure 4 - Layer 2

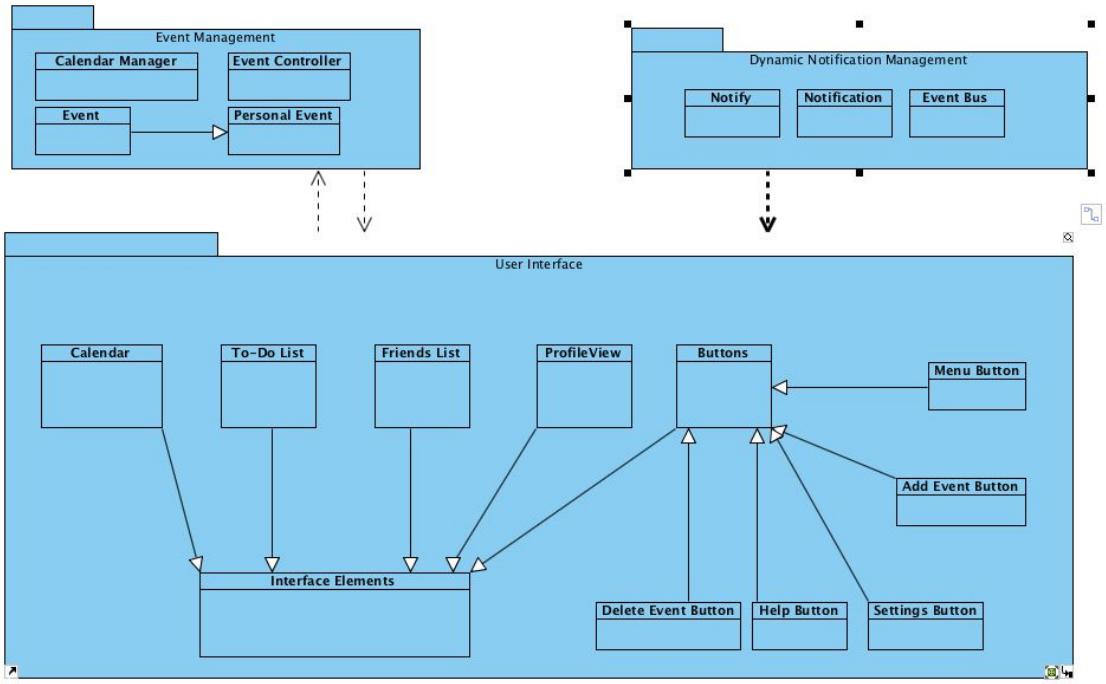

Figure 5 - Layer 3

The authorization subsystem collets user info which is then checked against a central dataholder in database to match the infos and allow the user to access his own profile. The user interface subsystem contains a javaFX API calendar and GUI material that are used to display events and to-do-list. The event manager subsystem is the controller section for creating and displaying all types of personal events. This section contains Calendar manager as a controller to Interface Elements in User Interface and Event Controller to control personal event objects and personal event model object. The smartEvent Manager is a subsystem to Event Manager which is responsible for creating and controlling smart events. However the displaying of this smart events is still done by calendar manager. The Profile Manager subsystem contain User Profile Controller which control user profile information and Friends Controller controls friend lists and creates new friends. The data management section is responsible for storing all the model objects and interacting with the database. It contains 3 data holder classes for storing event objects, profile objects and notification objects. The communication helper class allows for easy

communication between between holder objects and the database. The database interaction class is responsible for establishing networking with database. The system only contain a subsystem named Dynamic Notification Management which contains the Event Bus which has a active connection with the database and communicates on friends requests and smartEvent requests with other other users. If any friend request is gets accepted then Event Bus gives a pop-up notification and sends message to friends controller to add that user to friend list.

2.2 Model View Controller

Model View Controller is a architectural style for programming. Model View Controller(MVC) uses the idea of classifying the classes of the system into three components: model, view, and controller. In the system, we determined that are models. Those subsystems of the system will only be accessed and controlled by the .. classes. Other subsystems which are.. are responsible for providing the interaction between the user and the system, and the database. Those subsystems are view subsystems. Using this architecture style made our subsystems more dependent on each other and made the system more efficient. Therefore, using MVC is a good choice for this system.

2.3 Hardware / Software Mapping

Since our system is developed by Java, it needs Java Runtime Environment to be executed. For hardware requirements, the system needs a keyboard and a mouse to be controlled by user and also for interaction between them. For graphical part, Java graphics do not need too much GPU and it is compatible for majority of the personal computer systems.

2.4 Persistent Data Management

Most of the data is stored in the database. However, if the user is already logged in by using internet connection from a device, the user will be able to use the application's some of features like adding personal event offline. To acquire this feature, the system uses data management subsystem to store some of the data in device memory like saving events that are already in the calendar.

2.5 Access Control and Security

Users needs to register or login for the usage of the system. By asking two times for users' new password eliminates the problem of creating an password that a user does not know. Also a user needs to enter name, surname and valid email address to enroll to the system correctly. For login part, unique username and password are needed to access the profile and personal calendar. Users' passwords will be matched by md5 encoding to be secured.

2.6 Boundary Conditions

Initialization

Tempo is a click and run application for the personal computers that have java runtime environment. Therefore, there won't be a install file for Tempo. Tempe is a application that is an executable .jar file.

Disconnection

Tempo is a system that works with internet connection. In case of disconnection, many features of the system becomes not usable due to not connecting with the database.

Termination

Tempo can be terminated or closed by clicking the exit button on the user interface all the times. When a users clicks exit(X) button, the system will disconnect itself from the database. However, this termination will not log the user out of the system. The user will be able to use the system if there is no internet connection as an offline user.

Error

The system will give an error if the file or the saved data in the database are corrupted and will delete its contents.

3. Subsystem Services

In this section, the design patterns, the detailed design of the class diagram, and the explanation of the classes can be seen. These features many basic functions and how classes work.

3.1 Design Patterns

3.2 Detailed Class Diagram

3.3 Class Interfaces

3.3.1 Authorization Subsystem Interface

3.3.2 Data Management Subsystem Interface

3.3.3 Event Management Subsystem Interface

3.3.4 User Interface Subsystem Interface

3.3.5 Profile Management Subsystem Interface

3.3.6 SmartEvent Management Subsystem Interface

3.3.7 Dynamic Notification Management Subsystem Interface

3.4 Descriptions of Interaction Between Classes