**StudyFlow**

**Software Requirements Specification**

**09/04/2023**

**CSE3044 – Software Engineering Term Project**

**Name - Surname Number**

**1-) Eren ÇETİN 150120008**

**2-) Semih BAĞ 150120070**

**3-) Zeynep YILMAZ 150120066**

**4-) Hakkı KOKUR 150120033**

Prepared for

CSE3044 Software Engineering Term Project

[**1. INTRODUCTION 3**](#_1t3h5sf)

[1.1 Purpose 3](#_4d34og8)

[1.2 Scope 4](#_2s8eyo1)

[1.2.1 Objective 4](#_17dp8vu)

[1.2.2 Study Area 4](#_3rdcrjn)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_26in1rg)

[1.4 References 6](#_lnxbz9)

[1.5 Overview 7](#_35nkun2)

[**2. GENERAL DESCRİPTİON 7**](#_44sinio)

[2.1 Product Perspective 7](#_2jxsxqh)

[2.2 Product Functions 7](#_z337ya)

[2.3 User Characteristics 8](#_3j2qqm3)

[2.4 General Constraints 9](#_1y810tw)

[2.5 Assumptions and Dependencies 9](#_4i7ojhp)

[**3. SPECIFIC REQUIREMENTS 10**](#_2xcytpi)

[**3.1 Functional Requirements 10**](#)

[3.2 Non-Functional Requirements 14](#_1ci93xb)

[3.2.1 Performance 15](#_3whwml4)

[3.2.2 Reliability 15](#_2bn6wsx)

[3.2.3 Availability 15](#_qsh70q)

[3.2.4 Security 15](#_3as4poj)

[3.2.5 Maintainability 15](#_1pxezwc)

[3.2.6 Portability 15](#_2p2csry)

[3.3 Inverse Requirements 16](#_147n2zr)

[3.4 Logical Database Requirements 16](#_3o7alnk)

[**4.UML DIAGRAMS 17**](#)

[4.1 Use Cases 17](#_np7l5x3ehkno)

[4.1.1 Use Case #1 17](#_2g1rro23sta)

[4.2 Class Diagram 18](#_u4oo5zjpa2az)

[4.3 Design Sequence Diagrams 19](#_8oewejaz22z1)

[4.3.1 Swap Card 19](#_5640pm1ief2u)

[4.3.2 Swap Exercise 20](#_z4i7asf36k7y)

[4.3.3 Focus Activity 21](#_jrb65r4e1toc)

[4.3.4 Planning 22](#_hheicvf5tbty)

[**5. Work-Sharing 23**](#_l27jjic3x86)

# 

# 1. INTRODUCTION

StudyFlow application, developed by us to enhance the learning experience of students and individuals of all ages, aims to make the learning process more effective and efficient in today's fast-paced lifestyle. Our application offers various features to help users better track themselves and manage their learning processes more effectively.

Our goal is to facilitate users' learning processes and contribute to their success. In the face of the challenges and time constraints brought by the modern world, everyone needs to optimize their learning process. At this point, the tools and features offered by StudyFlow aim to make users' learning experiences more effective and efficient, ultimately facilitating real change and success in their lives.

The scope of our project includes the design of the application's features and user interface. Our aim is to provide a simple and understandable interface that users of all levels can comfortably use. In doing so, we aim to provide convenience for every type of user and make the learning process more enjoyable.

As for methodology, we adopt a continuous improvement approach based on user feedback. By taking into account users' requests and needs, we constantly update the application. Thus, we aim to continuously improve the user experience and make the application more effective.

The core process of the application is to enable users to track their study sessions and facilitate monitoring of their learning progress. Features such as the Pomodoro technique, saving and retesting errors, phone usage analysis, and focus times are designed to support this process and help users achieve success.

Our expectation is to help users achieve real and lasting success in their lives. The tools and features offered by StudyFlow will contribute to making users' learning processes more effective and reaching their goals. By continuously updating the application based on user feedback, we aim to provide the best experience for our users.

## 1.1 Purpose

StudyFlow application is a project aimed at improving the learning experience of students and individuals from all age groups in general. Our project's objective is to provide a tool for students and other users to optimize their learning processes and become more successful, addressing the challenges brought by modern life. Developed as a mobile application, StudyFlow encompasses features designed to track study periods, identify errors, and enhance concentration skills. However, the scope of our project is limited, as it does not offer a solution to completely transform students' learning processes. The target audience of our project includes not only students but also individuals of all ages and professions. The user base of the application encompasses professionals with busy work schedules, students preparing for exams, and anyone seeking to improve their learning process. Among the benefits StudyFlow offers to users are the development of more efficient and focused study habits, monitoring and analysis of learning progress, identification and resolution of errors, enhancement of time management skills, and overall improvement of the learning experience. Our project's main objective is to assist users in optimizing their learning processes. Key success criteria include user feedback, frequency of application usage, reduction in error rates, and user satisfaction. Possible risks our project may encounter include technical issues, changing user demands, and increased competition. To address these risks, strategies for project management and monitoring user feedback are continuously being developed.

## 1.2 Scope

### 1.2.1 Objective

Our main objective is to develop an application that assists users in organizing their daily lives and becoming more efficient."

### 1.2.2 Study Area

The focus of our project encompasses various activities that can aid in organizing the daily lives of individuals in general. The StudyFlow application aims to address users' needs in areas such as time management, focus skills, healthy habits, and productivity. The application's usage spans across a wide range of people with any lifestyle or profession, aiming to improve users' daily lives.

## 1.3 Definitions, Acronyms, and Abbreviations

| *Term* | *Definition* |
| --- | --- |
| HomePageActivity | This class manages the main screen of the application. It acts as a primary container for other tabs and functionalities. It is the first screen the user encounters when launching the application. It loads other tab classes, enables switching between them, and provides basic navigation for the application. |
| ToDoActivity | This class manages and displays the user's to-do list. It allows the user to add tasks, delete them, mark them as completed, and edit them. It performs these operations using the TodoController class. It provides the user with the ability to view and manage their to-do list. |
| FlashMindActivity | This class manages and displays the user's learning activities. It presents the user with "tags" representing subjects or activities. The user can swipe these tags left or right to select an action. Swiping right opens a tab where all cards within that tag can be viewed. Swiping left opens another tab to solve saved questions related to that tag. On the Flashmind page, you can also see the number of questions entered specifically for each tag. |
| Pomodoro Technique | The Pomodoro technique is a time management and productivity method that relies on short bursts of focused work followed by regular breaks. Typically, it involves a 25-minute work period (known as a "Pomodoro") followed by a short break. This cycle is repeated, and after completing four Pomodoro work cycles, a longer break is taken to rest and recharge. |
| FocusActivity | This page provides an interface for implementing the Pomodoro technique. Users can start, pause, or cancel the Pomodoro session. It helps users track their focus periods and increase productivity. |
| PlanningActivity | This page provides a calendar interface for users to manage their plans. Users can input tasks for specific dates and edit or delete these plans as needed. Additionally, it allows users to remove unwanted tasks from the calendar. |
| TagActivity | This page provides an interface for managing tags used in the application. When users start a new activity and any necessary tags are missing, they are redirected to this page. From here, users can perform operations such as adding and removing tags. |
| AnalysisActivity | This page presents the results of the user's actions within the application (such as Pomodoro durations, correct/incorrect questions in solved exercises) in graphical and textual format. It allows the user to view the analyses either specifically for certain tags or encompassing all tags. |
| ExerciseActivity | This page provides an interface for reviewing questions entered specifically for certain tags. Users can view questions based on the selected tag and only check their answers. |
| Tag | "Tag" is a descriptor representing a class, an activity, or a subject within the application. For example, activities such as "swimming", "math class", or "knitting" can be considered as tags. Each tag can represent the amount of time spent on it or other information associated with it. Tags help users organize their activities, analyze their progress, and focus on specific areas of interest. |

## 1.4 References

This recommended practice shall be used in conjunction with the following publications.

“Develop for Android.” *Android Developers*, https://developer.android.com/develop. Accessed 9 April 2024.

“Kotlin ile Android Mobil Uygulama Geliştirme Temel Seviye.” *BTK Akademi*, https://www.btkakademi.gov.tr/portal/course/kotlin-ile-android-mobil-uygulama-gelistirme-temel-seviye-10274. Accessed 9 April 2024.

*25:00 - Time to focus!*, https://pomofocus.io/. Accessed 9 April 2024.

“Pomodoro Technique.” *Wikipedia*, https://en.wikipedia.org/wiki/Pomodoro\_Technique. Accessed 9 April 2024.

Tümer, Borahan. *Lecture Slides*.

Kaya, Mehmet. *Lab Slides*.

## 1.5 Overview

StudyFlow is a mobile application designed to enhance the learning experience for students and individuals of all ages. The app assists users in tracking their study sessions, identifying mistakes, and improving their focus skills. Furthermore, it offers benefits such as analyzing learning progress, enhancing time management, and overall improvement of the learning experience. StudyFlow is tailored to help students overcome the challenges of modern life and achieve greater success in their learning endeavors.

# 

# 2. GENERAL DESCRİPTİON

Our product is design for tracking and carry out tasks which the user is working on more efficiently and see their analysis while doing that and improve the productivity.

## 2.1 Product Perspective

**·** When the user chooses a planning feature in the application, the calendar is shown in the screen and the user can choose a planning time which can be daily, weekly etc. and the user can select a day from that calendar and write plans into that day.

**·** When the user focusing mode which is also called as pomodoro. Pomodoro technique is an emerging technique for working more efficiently. There are time limits for both focusing and break time. Focusing time is twenty-five minutes and five minutes for break times. That limits are default arranged by pomodoro technique, but it can be change to the anything by the user.

**·** In our software product, we need to store some information. For that purpose, we use SQlite product. With the SQlite product, we store information and we can get that information efficiently when we need to.

## 2.2 Product Functions

There are some major functionalities that the software will perfom:

One of them is time-arrangement or the other name of this is focusing mode. That function provides user to set a spesific time to focus his/her work and time for break time after focusing session. It increases efficieny of doing work with a time.

The other one is determining and planning what to do within a certain day. The user can plan what he/she will make within a certain day in a spesific month and year. This functionally provides a track of works which must be done within a certain period of time and arrange their works according to them.

One other functionally of that software product is to do list. The user can write and manage the planning of future unknown events or some importing events here. The user can also tick that events here that indicates that is done or delete that from the to-do list.

The another functionally of our software is reminder feature. The user is reminded for a certain period of time so that he/she can learn something he/she does not know or something he wants to reinforce well. This period is determined by given answer to the given topic in the reminder part. It depends on True or False respons to the given answer by user.

One of them is loading or deleting information for the reminder feature. For reminder feature, there must be information on the app. The user can upload an image with additional comment or just write anything the user wants with their answers as a image or text. For deleting information, just select the item which is wanted to delete and delete selected item from remainder list.

The last function of our product is analysis. According information of successfull categorized sessions in pomodro or focus mode, An analysis is made based on that information and it is presented to the user to track his/her activities in pomodro or focus mode.

| **S. No.** | **Particulars** |
| --- | --- |
| 1 | Focusing mode(Pomodoro) |
| 2 | To-Do-List |
| 3 | Planning Event |
| 4 | Load Visual Information for Reminder ( Flashmind) |
| 5 | Load Text Information for Reminder (Flashmind) |
| 6 | Delete Information in Flashmind |
| 7 | Reminding Information |
| 8 | Analysis |

## 2.3 User Characteristics

The first user type that can be use our software is students generally attending high-school or university. For high-school students, they are generally using for exams which is need to go to university. They can track their informations about lectures and make their plans for that and improve their efficiency while working. For university students, they are also using for their lectures and projects which they are working on school for efficient time management.

The other type of user is white-collar workers. The white-collar workers works on project at their works and it plays an important role for their works. So it needs a planning and using time efficiently because they have limited time for their projects. With our product, they can arrange their time and works according to time efficiently.

These people can use that software product frequently but other people can also use that software product.

## 2.4 General Constraints

While developing the app, the pages of app has to be basic and understandable from the user because If we design it in a complicated way, after a while the user will not understand what he/she is doing and will cause him/her to quit the application.

The personal and user information must be secured in a good way. These datas shall not fall into hand of other people.

The algorithms that form the backend of our software product must work as quickly or optimally as possible. Because If our product works correctly but remains slow, it will cause the user not to use our product after a while.

## 2.5 Assumptions and Dependencies

**·** Our application is developed in the android-based system. If there is a different operating system from the android system, Our application doesn’t work. So, the requirements must be changed according to operating system.

**·** Our software product is only run at cell-phones. If another source other than the phone is used to open the application, It does not work. So, the requirements must be changed according to where we want to open the application.

# 3. SPECIFIC REQUIREMENTS

## 3.1 Functional Requirements

| ***S. No.*** | ***Features*** | ***Functionality*** | ***Descriptions***  ***(Including Sub-Functionality)*** |
| --- | --- | --- | --- |
| 1 | Home Page | When the user comes to the home page, he/she should be able to manage all his/her transactions from there. | The user should be able to go to the FlashMind page with a single click on the home page. The user will keep what he has learned fresh by doing exercises on this page. |
| 2 | The user should be able to go to the To Do page with a single click on the home page. The user will be able to easily manage the tasks he needs to do on this page. |
| 3 | The user should be able to go to the Focus page with a single click on the home page. On this page, the user will be able to initiate pomodoros to focus by customizing their own settings and wishes. |
| 4 | The user should be able to go to the Analysis page with a single click on the home page. On this page, the user will be able to see analyzes and reports prepared specifically for him. |
| 5 | Tag | On this page, the user will be able to manage the tags he will use while focusing or doing reminder exercises. | The user can sort and view all tags. |
| 6 | The user can add a new tag according to his/her wishes |
| 7 | The user can delete a tag from the list according to his/her wishes. |
| 8 | User can color a tag according to his/her wishes |
| 9 | The user can remove the tag by clicking on the cross icon to the right of the tag. |
| 10 | Analysis | On this page, the user should be able to see detailed reports and analyzes of the pomodoros and card exercises she/he has done so far. | The user should be able to see a small informative text at the beginning of the analysis page. |
| 11 | The user should be able to see detailed analyzes according to the tags he/she has previously set on the analysis page. |
| 12 | In order to make it easier to follow the reports prepared by the system, the user should be able to see the listed reports and analyzes by scrolling down the page on the analysis page. Instead of clicking for each report, you should be able to access the reports by scrolling down. |
| 13 | On the analysis page, the user can choose to benefit from visualization with various graphs such as line charts, column charts. |
| 14 | Planning | When the user enters the planning page, the user can see and manage the plans he has previously created. | When the user first logs in to the planning page, a calendar is printed on the screen. |
| 15 | When the user first logs in to the planning page, a calendar is printed on the screen, and by default, the date information for that day is selected. |
| 16 | The user can select the desired date on the calendar. |
| 17 | The user should be able to change the month information on the calendar by swiping the screen left and right on the calendar. |
| 18 | When the user selects a desired day on the calendar, all plans for that day should be listed under the calendar area, where data such as text and title can be seen. |
| 19 | If the screen size is exceeded while printing the plans for the day the user has selected, the user should be able to see all the plans by scrolling down the screen. |
| 20 | If there are no plans for the selected day, an explanation text should be printed where the plans are listed and printed on the screen. |
| 21 | The user should be able to choose a day of his choice and easily cancel the plan he had previously planned for that day. This can be easily canceled by clicking on the cross icon on the right side of the listed plans. |
| 22 | The user can add more than one schedule for a selected day. |
| 23 | To add a plan, the user must first select a day on the calendar. After the plans for that day, if any, are printed on the screen, the user can click on the add plan button at the bottom of the screen. |
| 24 | In the window that opens when adding a plan, the user can add any text entry he wants to the system as a plan, as long as it does not exceed the limits set by the system. |
| 25 | Focus | The user should be able to use this page to focus. On this page, the user can focus with the Pomodoro technique, with settings that he can customize. In this way, it is aimed to do the job he/she is focusing on more efficiently. | When the user enters the focus page, the duration of focus must be selected in minutes. |
| 26 | When the user enters the focus page, the focus time should be automatically determined as 25 minutes as the initial value. Afterwards, the user should be able to adjust it according to the user's wishes. |
| 27 | The user should be able to choose the break time between focus periods according to his/her own wishes. At the same time, this break time should automatically come with an initial value of 5 minutes. |
| 28 | The user should be able to choose which tag to focus on during the focus period. In this way, he/she should be able to see which tag he/she spent more time on and focused more on from the analysis page. |
| 29 | After adjusting the Pomodoro settings to their liking, the user can start the focus period by pressing the start button. |
| 30 | After the user starts the focus period, there should be a counter to track the remaining time on the screen. |
| 31 | After the focus counter is started by the system by pressing the start button, the cancel button becomes active on the screen. |
| 32 | The user must be able to cancel the focus before the set time expires. |
| 33 | The user should not be able to press the cancel button before starting to focus. |
| 34 | After the user successfully completes the focus period, he/she should be able to start the break with another click to start the break period. |
| 35 | Once the user has successfully completed the focus period, no manual interaction should be required for this work to be included in the analysis and report. Everything should be handled by the system. |
| 36 | FlashMind Page | The user can see all the tags on this page and add cards to that tag, or if there are cards in those tags that need to be repeated, he can repeat those cards. | When the user logs in to the FlashMind page, he should be able to see all the tags he has created so far in a list. |
| 37 | The user should be able to switch to the exercise page by pressing and holding one of the listed stones and sliding it to the right. |
| 38 | The user should be able to switch to the My Cards page by pressing and holding one of the listed tags and swiping it to the left. |
| 39 | Cards | It is a reminder card system based on the user's forgetting curve, so that the valuable information the user has experienced and learned is not forgotten. The user should be able to manage these cards and the system should be able to provide question and answer interactions to the user through the cards. | The user should be able to see all cards belonging to the tags he has chosen. |
| 40 | The user should be able to see only the titles of the listed cards. |
| 41 | Users who want to see the data on the front and back of the card should be able to click on the card to go to a page that prints the card's data on the entire screen. This should be done by simply clicking on the card. |
| 42 | While the user examines the data of the card in a way that covers the entire page, when he wants to see the data on the back side, he should be able to see the data on the other side by clicking anywhere on the screen. |
| 43 | The user can remove any card from the system by pressing the cross icon next to the card among the listed cards. |
| 44 | The user can add a card containing text to the system. |
| 45 | The user can add a card containing a picture to the system. |
| 46 | The user can remove any card from the system by pressing the cross icon next to the card among the listed cards. |
| 47 | While the user sees the cards in a list, he/she should be able to access all cards by scrolling down the screen. |
| 48 | When the user wants to add a card, they should be able to click on the add card button at the bottom of the screen. |
| 49 | Exercises | The user must be able to exercise with the cards determined by the system. These cards should be updated based on the answers given by the user in the previous exercise. The system will do this update automatically. | When the user logs in to the exercise page, the title and front page content of the card should be printed on the screen. The user should not take any additional action to see this data. |
| 50 | The user must answer the question on the card by himself. It should not make any choice or text entry. |
| 51 | The user must be able to click on the show answer button to see the answer to the card. |
| 52 | The user must be able to click on the show answer button to see the answer to the card. |
| 53 | If the user thinks he answered the question correctly, he should be able to skip the question by pressing the correct button. |
| 54 | If the user thinks he answered the question incorrectly, he should be able to skip the question by pressing the incorrect button. |
| 55 | If the user thinks he answered the question incorrectly, he should be able to skip the question by pressing the incorrect button. |

## 3.2 Non-Functional Requirements

The application can deliver a reliable and user-friendly experience that empowers users to effectively manage their time, organize their tasks, and enhance their learning through flashcards. These NFRs provide a strong foundation for developing a comprehensive productivity tool that caters to the diverse needs of users

### 3.2.1 Performance

The pomodoro timer functionality shall exhibit minimal lag and ensure a smooth user experience. The application should respond to user actions, such as starting, pausing, and resetting the timer, within 1 second. Additionally, the visual and auditory cues marking the beginning and end of intervals should trigger precisely at the designated times, with a tolerance of no more than one minute. This performance requirement applies to all supported devices and operating systems, even under moderate system load, to guarantee an uninterrupted flow during focused work sessions and breaks.

The application should maintain consistent performance even during peak usage periods. This ensures users can add tasks, set timers, and access flashcards seamlessly.

### 3.2.2 Reliability

The application should implement robust data recovery mechanisms to safeguard user information in case of unexpected shutdowns or device malfunctions. This ensures minimal data loss and protects critical details like scheduled events, timers and remainders.

### 3.2.3 Availability

Upon restart after a crash or bug, the application should automatically recover the most recent data. This allows users to resume their tasks, pick up where they left off with their schedules, and continue studying with their flashcards without manually re-entering lost information. By implementing these data persistence and recovery mechanisms, the application guarantees user productivity and minimizes disruptions caused by unforeseen issues.

### 3.2.4 Security

The application should undergo regular security testing and patching to identify and address potential vulnerabilities promptly. This proactive approach minimizes the risk of cyberattacks and safeguards user data from exploitation. By adhering to these security NFRs, the application fosters user trust and empowers them to confidently manage their tasks, schedules, and learning without compromising their privacy

### 3.2.5 Maintainability

Well-documented code with comments and explanations will enhance understanding for developers who maintain the application in the future. This reduces time spent deciphering the code's logic and streamlines the maintenance process

The application should be built using a modular architecture, where functionalities like the pomodoro timer, scheduler, and flashcards are separate modules. This simplifies future modifications and reduces the risk of unintended consequences when changing one section of the code.

#### 

### 3.2.6 Portability

The application should adhere to relevant industry standards and best practices for mobile development. This promotes code clarity and maintainability, making it easier to port the application to new platforms that follow similar standards. By following these portability NFRs, the application gains flexibility for future expansion and reaches a wider user base across different mobile ecosystems

## 3.3 Inverse Requirements

* **Unnecessary Complexity:** The application should avoid overly complex interfaces or functionalities that overwhelm users. A clear and intuitive design is essential for users to efficiently manage their tasks, schedules, and flashcards.
* **Data Leakage:** The application should not collect or store any user data beyond what's necessary for core functionalities. This protects user privacy and avoids potential security risks associated with unnecessary data collection.
* **Battery Drain:** The application should be optimized for efficient battery usage. Excessive background processes or resource-intensive features can quickly drain battery life.
* **Internet Reliance :** If the core functionalities (pomodoro timer, scheduler, flashcards) can function offline, the application should strive to minimize its dependence on an internet connection. This ensures users can be productive even in situations with limited or no internet access.

By adhering to these inverse requirements, the application prioritizes a user-friendly experience, minimizes distractions, and safeguards user privacy. This focus on what the application should not do complements the traditional NFRs, ultimately leading to a well-rounded and effective productivity tool.

## 3.4 Logical Database Requirements

The success of this productivity application hinges on a robust and secure local database to manage user information. Since the application utilizes SQLite as its embedded database engine, the following non-functional requirements (NFRs) are crucial for ensuring data integrity and accessibility.

The local SQLite database should persistently store all user-generated data associated with the application's core functionalities. This includes information related to pomodoro timers (duration, history), schedules (events, tasks), and flashcards (decks, content, progress).By persisting this data, the application ensures users don't lose their work due to application crashes, device restarts, or unforeseen shutdowns. Users can confidently rely on the application to retain their progress and pick up where they left off.

# 4.UML DIAGRAMS

## 4.1 Use Cases

### 4.1.1 Use Case #1

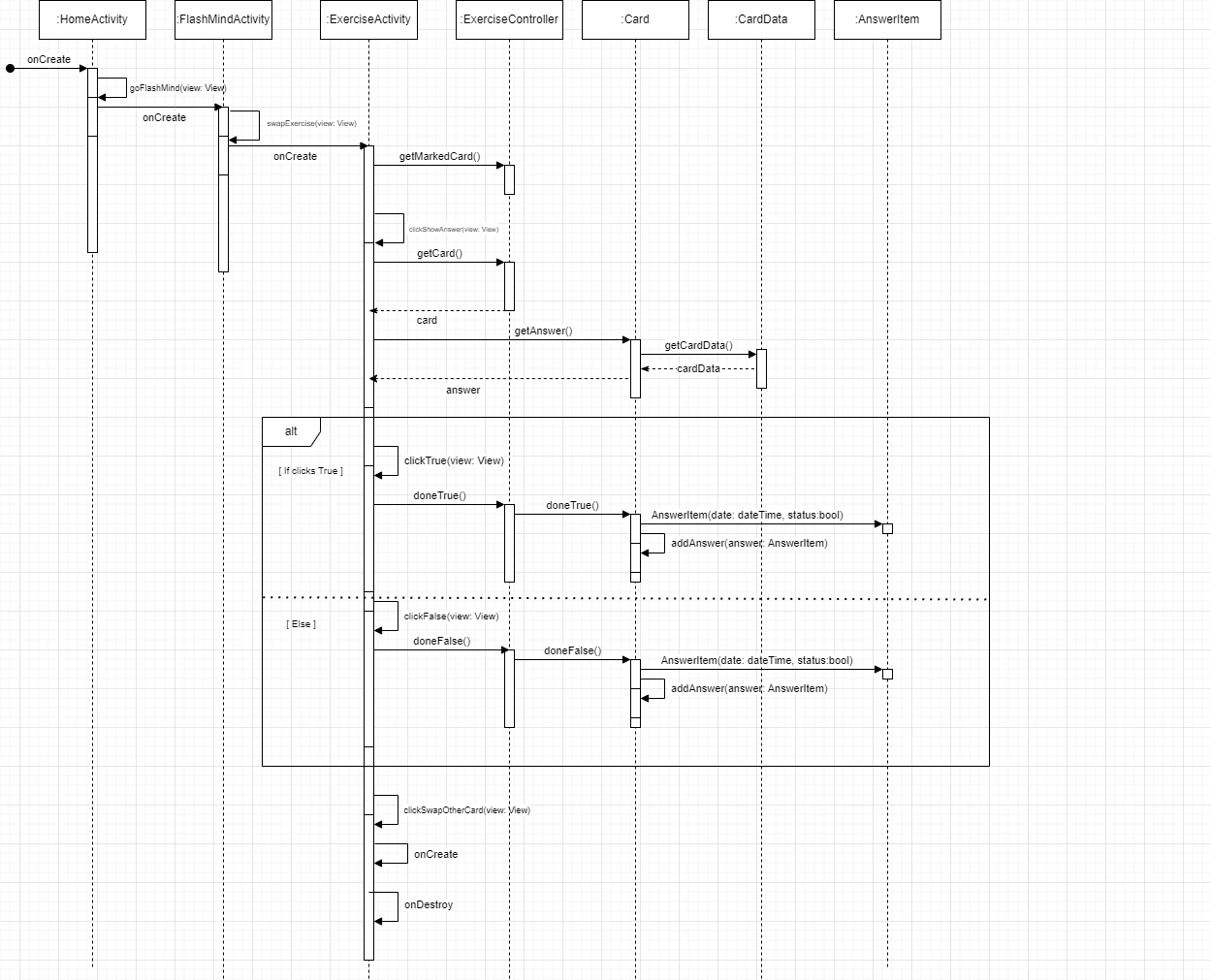
## 4.2 Class Diagram

## 4.3 Design Sequence Diagrams

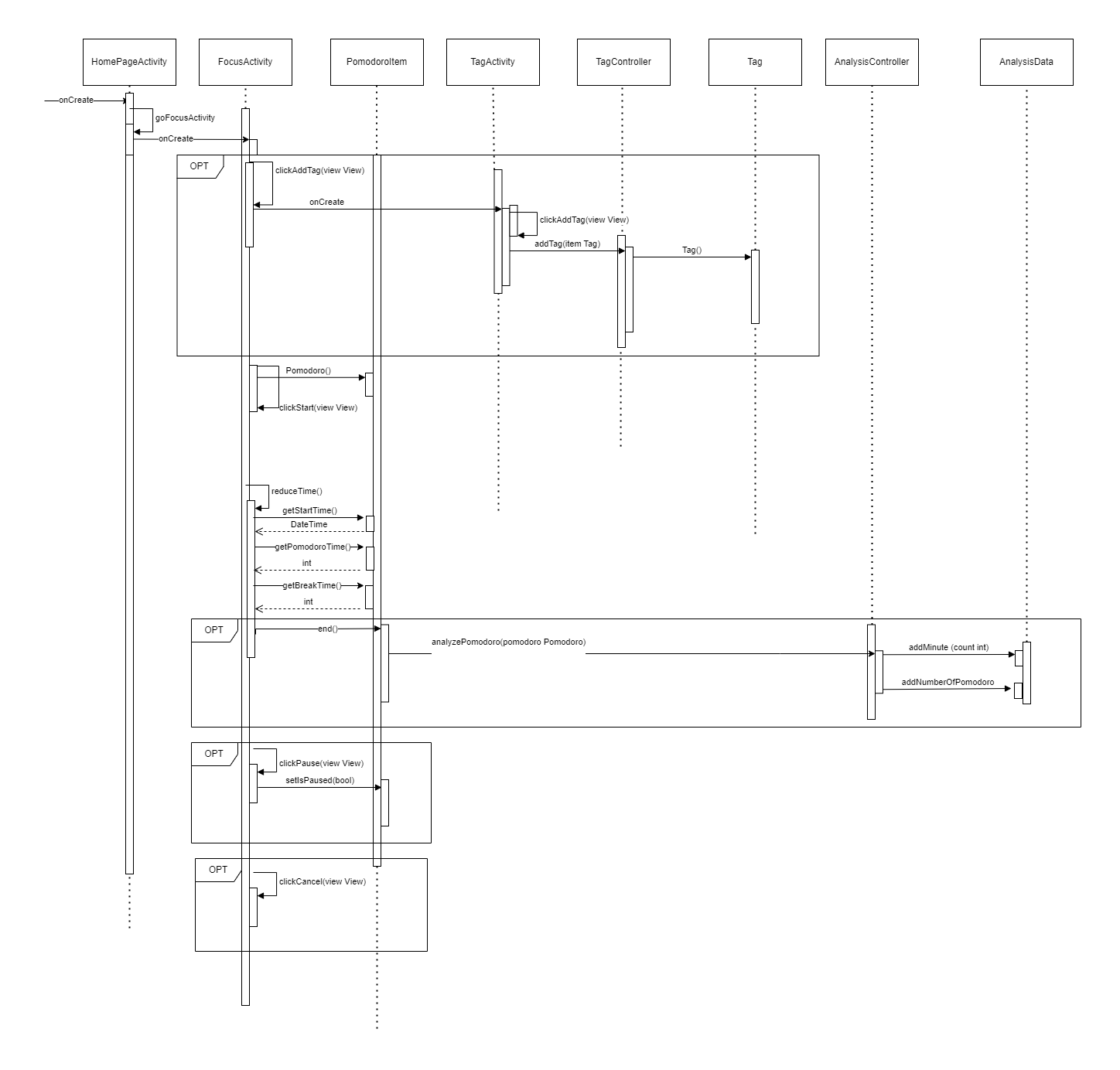
### 4.3.1 Swap Card

### 

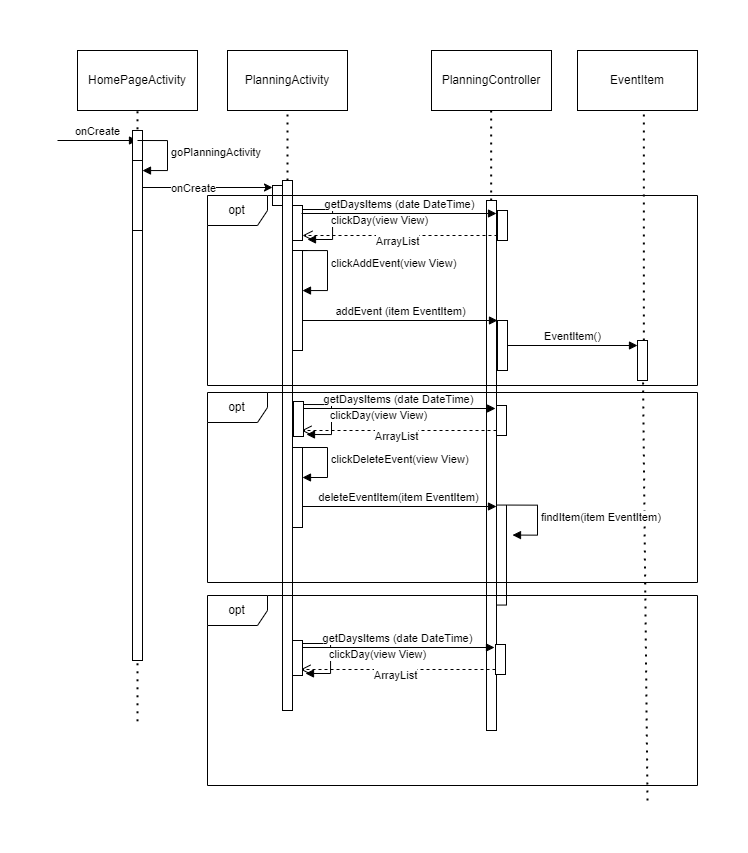
### 4.3.2 Swap Exercise



### 4.3.3 Focus Activity



### 4.3.4 Planning



# 

# 5. Work-Sharing

| Section 1 | Zeynep Yılmaz |
| --- | --- |
| Section 2 | Eren Çetin |
| Section 3 | Semih Bağ, Hakkı Kokur |
| Section 4 | Eren Çetin, Zeynep Yılmaz, Semih Bağ, Hakkı Kokur |