StuderaMera

Requirement and analysis document

Izabell Arvidsson, Ida Dahl, Julia Jönmark, Hanna Söderström

September 2020

Contents

1 Introduction

1.1 Purpose of application

The project aims to create a computer based application which intends to help students manage their time, with the help of a schedule, to-do list, and a timer. In addition, one of the core features is the digital flower which will grow to reflect the students progress and mark their achievements.

1.2 General characteristics of application

The application is adapted to desktops and will be sized to fit most displays.

The application will have three main pages. The first main page is where the users can see their calendar and if they choose, they can change it. They will be able to add, remove or rearrange appointments.

The second main page is where the users can set how long they will study for and how many breaks they need. There will be a timer which counts down to the next break and a digital flower which will grow at the same speed as the user is studying. If the user chooses to close the application during the ongoing timer, the digital flower dies. In addition there will also be a tip-button on this page which the user can click on and read about different study methods.

The third main page is for organizing tasks that need to be completed. It is compromised of to-do lists where you can can write up tasks and check them of as they are completed.

1.3 Scope of application

The application will not allow the users to share its calendar or to-do list with others or to other devices. The user will not be able to import calendars or to-do lists from outside actors. The application doesn't support multiple users.

1.4 Objectives and success criteria of the project

- It should be possible to create a calendar where the user can colour code, set a date and time, and give a description on the activity.
- It should be possible to create a list of tasks, where the user can specify the task, can organize the different tasks in separate maps and can check

off a task when it's done.

- The application should help a student to organize their life with the help of the calendar and to-do list. The application should help with a student's concentration during studying.
- The application should save the users data when it is closed.

2 Requirements

2.1 Functional requirements

The user should be able to:

1. Choose whether to start planning or studying
2. When entered the studying part of the application choose
   (a) The number of minutes to study
   (b) The number of minutes to rest
   (c) The number of repetitions
   (d) Read about tips for studying
   (e) Start the timer
3. When entered the planning part of the application choose
   (a) To see the overview of the the to do-list and the day's calendar
   (b) Enter the view over the calendar
      i. See an overview over the entire week
      ii. Switch between weeks
      iii. Add and remove from the calendar
   (c) Enter the view over the to do-list
      i. See the entire list
      ii. Tick off tasks on the list
      iii. Add and remove from the list
   (d) Add and remove from the list
4. The application should save the users data when it is closed.

2.2 Non-functional requirements

2.2.1 Usability

The application will have focus on usability. The application will assume that the user has prior knowledge with calendar and timer application, and the language in the application will be Swedish. To convey information to the user, the application will equally use text and graphics.

2.2.2 2.2.2 Reliability

N/A

2.2.3 2.2.3 Performance

The user should not notice any delay with the application, and any input the user makes should seem immediate.

2.2.4 Supportability

The program should be able to run on Windows, Linux and Mac platforms. The application will be built for desktops.

There will be automated tests for all parts that it can be applied to and manual tests for the parts that automated tests can not be applied to.

2.2.5 Implementation

The application will use the Java environment. In order to run the application the user will need to have a program that can run a java file.

2.2.6 Packaging and installation

The application will consist of a executable file for running the program and all needed resources in a zip archive.

2.2.7 Legal

The application contains several images which are licensed under the Creative Commons license.

The name of the application might have legal issues but that will not be covered in this document.

2.3 Application models

2.3.1 Use case model

See appendix 1 for UML diagram of use cases.

2.3.2 Use cases priority

High priority:

- Start timer
- Stop timer
- Add Calendar event
- Add to-do list

Medium priority:

- Set rest timer
- Set study timer
- Set repetitions
- Edit calendar event
- Edit to-do list
- Ability to set color of calendar event

Low priority:

- Get study tips from light bulb
- Detail view of calendar events

2.3.3 Domain model

See appendix 3 for the analysis model.

2.3.4 User interface

The user interface is designed to help focus on tasks such as studying with a clean interface with few distractions when the study timer is running. It is also designed to aid the user to keep track of tasks and where to be and when with help of the calendar and to-do list. See appendix 4 for preliminary GUI

Appendix

Appendix 1 : Use case model

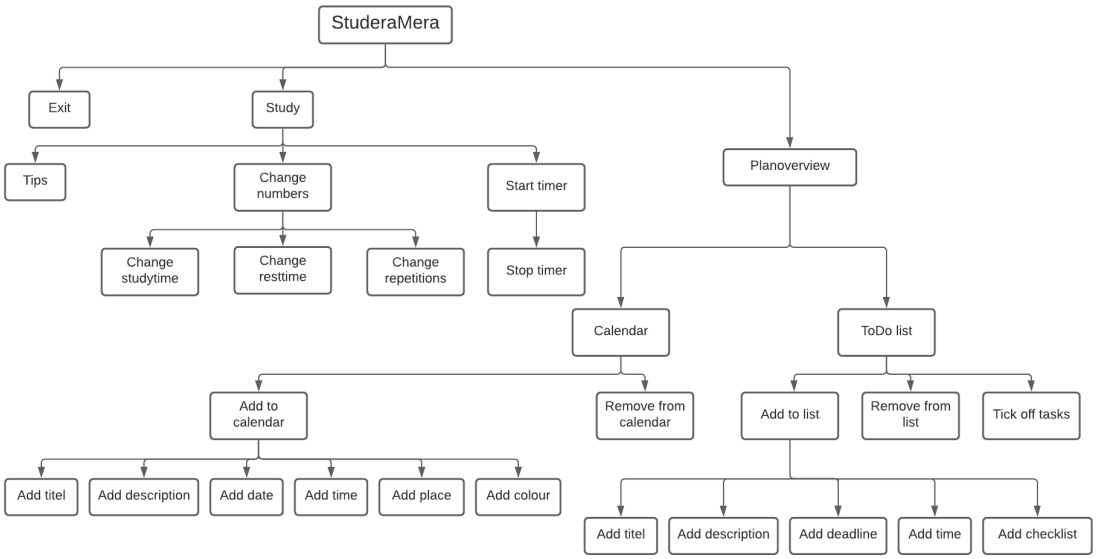

Figure 1: figure 1: UML of the use cases for StuderaMera

Appendix 3 : Domain model

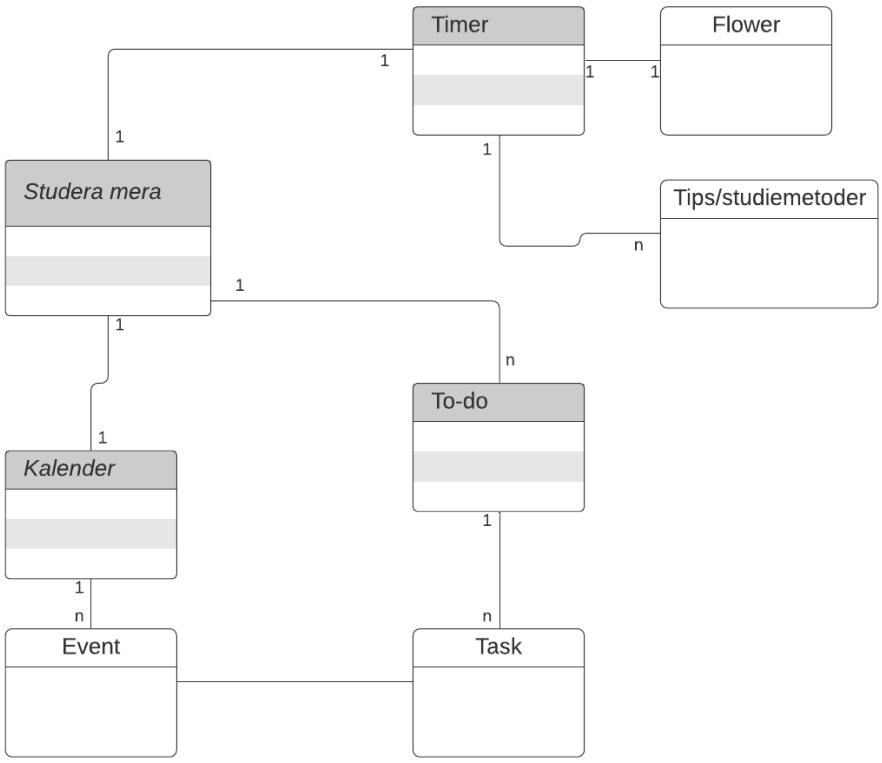

Figure 2: figure 1: Preliminary domain model for StuderaMera

Preliminary GUI

Figure 3: figure 3: Preliminary GUI for startpage

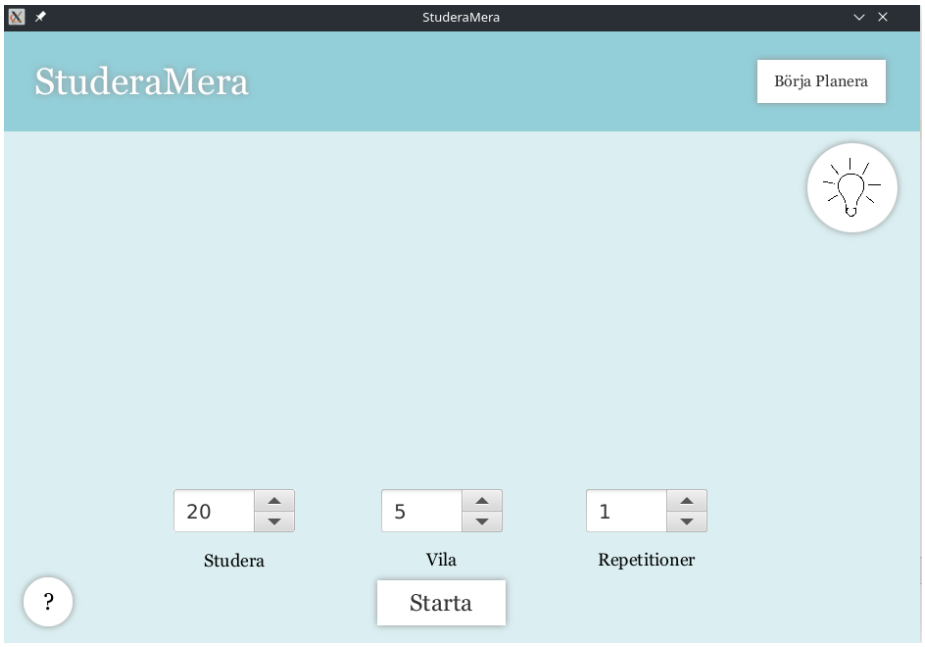

Figure 4: figure 4: Preliminary GUI for timer page

Figure 8: figure 1: Preliminary GUI for to-do lists

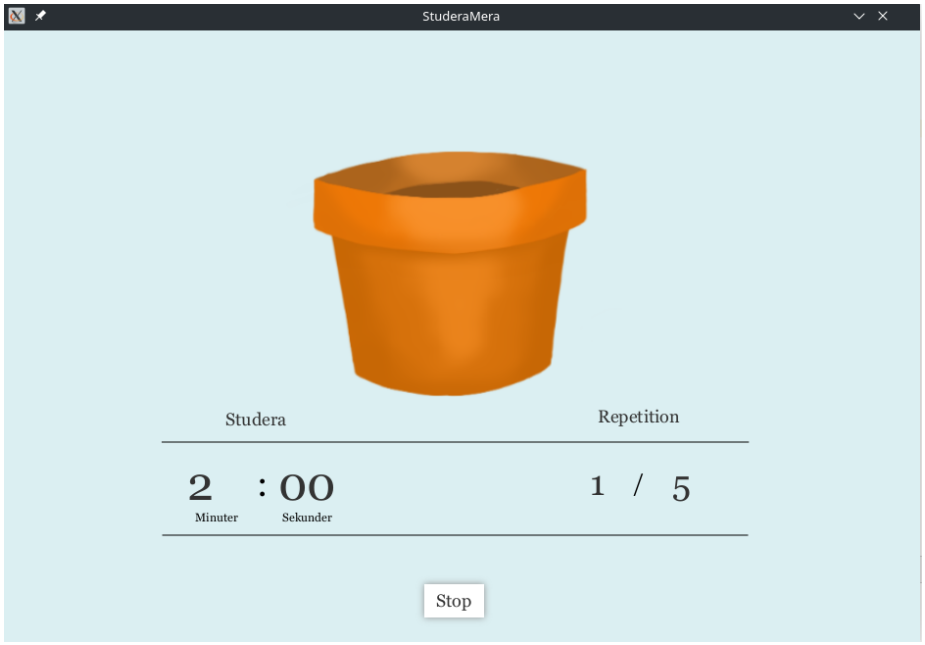

Figure 5: figure 5: Preliminary GUI started timer page

Figure 6: figure 6: Preliminary GUI planning page

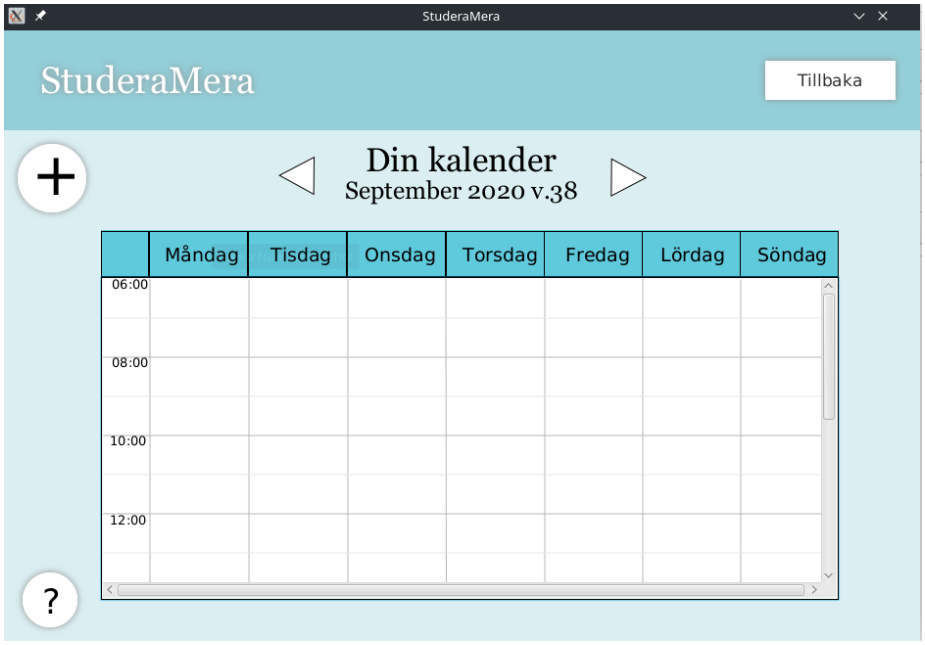

Figure 7: figure 1: Preliminary GUI for calendar page