پروژه طراحی نرم‌افزار

- **نام درس : اصول طراحی نرم‌افزار**
- **نام استاد مربوطه : دکتر حاجی‌اسمعیلی**
- **واحد دانشگاهی : تهران مرکزی**
- **ترم بهمن 1403**
- سوگل شریعت‌زاده

- Topic : software design principals
- Lecturer: dr.maryam hajiesmaeili
- University branch: Azad Islamic university-tehran central branch
- February 2024
- By: Sogol Shariatzadeh

Software name: TimeTact

✓ Tactfully handling time and tasks

- **Software description:**

Personal Time Management and Calendar Platform: Design a software platform that allows users to manage their time, schedule, and personal calendar. This platform can include features such as creating weekly and daily plans, announcing reminders and important information, sharing calendars with friends and colleagues, shared calendars for families and groups, and reporting on time spent on various activities. This platform can help users optimize their time, better plan daily activities, and help achieve personal and professional goals.

- **Special and distinctive feature of the software:**

As you know, there are similar software on the market, but what makes our software different is the way it punishes and encourages.

If the user does not complete the tasks entered in the program, the software accesses the user's account, which was filled with some money when the program was started, and deducts some money for each task not completed. In return, the software withdraws money and if the user completes the tasks within a certain period of time, it returns it to the user in the form of cinema tickets, amusement parks, etc.

VISION

•TimeTact is a planner and Daily Tasks software that falls into the Productivity product category. It is suitable for every class and individual who needs daily planning, including students, pupils, managers, etc., which helps you manage your time in the best possible way and complete your tasks effectively. This program can help you in planning and managing time to achieve greater productivity and complete your tasks in an organized and optimal manner. The reason for its production and its seriousness in punishing and encouraging goes back to today's lifestyle, which has a high rate of procrastination, forgetfulness, and so-called procrastination.

•TimeTact software can be useful for different age groups who are able to read and write and are allowed to use mobile phones. But the point that makes the development team limit the age range of this software to +15 is that not doing tasks in this software causes a loss of some capital, which of course is not significant for children and may cause problems with negligence. But in general, this program is suitable for people who need to plan their day accurately and are required to act on all of them, such as students, students, business managers, teachers, project managers, office workers, entrepreneurs, technical specialists, etc.

•But in addition, people who are lazy can use this software a lot because these people are regressing due to this personality trait and are not so productive.

•TimeTact is a program in the productivity program category. (organizer, planner, tasks, reminders)

•The compelling reason that the development team of this software has for using TimeTact is that this program puts its consumers in a fun situation and this can lead users step by step towards a more useful life.

•Unlike other software, this software puts money in the middle and is not satisfied with taking pictures and solving mathematical equations, and not doing tasks causes financial losses to the user, so that they are reminded that this procrastination is not the main loss of money, but in fact can harm the future and work life of the person.

•Our product's access to a user wallet in the category of this type of program and creating a platform for gifting various types of tickets and shopping vouchers are among its primary differences.

1. For students, managers, teachers, etc. (age group + 15)
2. Anyone who needs planning
3. TimeTact is a productivity software
4. Taking tasks seriously is a compelling reason to install the product and pay for it
5. Unlike similar software available in the market, the seriousness of completing tasks is very high
6. The feature of accessing the user account and the ability to give prizes distinguishes our product

REQUIREMENTS

1.The software will be able to create tasks, prioritize, remind, and categorize

2. The software will be able to withdraw money from a person's account

3.The security of the software will be high, and the authentication operation will always be performed at the highest speed when entering the program

4.The software's UI should be simple and at the same time attractive and easy to use

5.The data entered by the user is placed in a special database

6.The software update should be actively carried out so that bug notifications by users are at their lowest level

7.The software will allocate gifts to users if they use the program optimally

8.Considering the point of differentiation of the program compared to similar products, the support for this software should be dynamic and active

8.The product support unit should be formed

10. The security of the money remaining in the user's account should be ensured

11.The program should support three types of Gregorian, Solar, and Hijri calendars

12. Training on working with the program should be designed on a page of it

13.The software should support the world's modern languages and add them over time

14. There should be an option to completely delete the user account and an option to log out of it

15.There should be different themes for running the program

SCENARIO

•Scenario 1

Zahra, a 21-year-old dentistry student at Shahid Beheshti University, went to the university library after her afternoon class and wrote down on a piece of paper a few things she needed to do in the coming days, and also reviewed what she had done in her classes last week. She decided to model her class activities more in the classroom and interact more with the relevant teacher in order to learn the lesson better.

•Scenario 2

Sara opens her planner at the end of the week as her routine and first checks the status of her tasks and assignments. By evaluating what she has done so far and checking the completed tasks, she confirms her progress by ticking the tasks. Then, based on the checked progress and the remaining tasks, she reviews and revises her weekly plan for the next week.

•Scenario 3

Ali is an employee and writes his tasks on his to-do list every day. He divides the to-do list into It is divided into three parts, where his priority is important work from 10 am to 1 pm and then until 4 pm. After reviewing the work, if the work was done correctly and completely, he can invite himself to a favorite pastime after work to encourage himself. If he did not do his work properly, he should help with the housework more at night and not leave much time for his own entertainment.

•Scenario 4

A group of 4 friends decide to go on a trip for the summer vacation. The friends discussed the destinations they want to go to and they reached an agreement and concluded that they would go to Mazandaran and stay for 4 days, staying at the Jangi residence on the first day, on the second day by the beach, and on the third and fourth days they would be busy shopping and buying souvenirs.

USE CASE

Main:

Daily planning
Task prioritization
Task editing
Opening an account
Setting and tracking goals
Daily goal setting
Time management
Project management
Financial management
Task management
Task scheduling
Task reminders
Preparing a task list
Deadline tracking and monitoring
Task tracking and monitoring
Recording plans
Checking tasks
Viewing task lists
Accessing the calendar

Sub:

Support
Personal fulfillment and development
User encouragement
User punishment
Report writing
Trend analysis
Task sharing
Event recording and...
Personal coordination
Learning management
Change management
Assigning tags to tasks
Viewing task statistics and graphs
Extracting time and individual patterns
Adding explanations to tasks
Viewing task lists based on tags
Planning and management (events - occasions - travel - entertainment - health - contact - meetings, etc.)(

Use case document no.1

Use case name:	مشاهده لیست وظایف	
Scenario:	مشاهده انلاین لیست وظایف تشکیل شده	
Triggering event:	کاربر میخواهد لیستی که نوشته را مشاهده کند	
Brief scription:	کاربر به صفحه کاربری خود رفته و وارد قسمتی که وظایف را نوشته میشود تا ان هارا مشاهده کند	
Actors:	کاربر	
Related use case:	پیگیری و مانیتورینگ وظایف - تیک زدن تسک ها - تیک زدن تسک ها - ویرایش وظیفه	
Stakeholders:	UX designing – accounting – lists – viewing	
Preconditions:	کاربر باید برای مشاهده لیست وظایفش ازقبل حساب کاربری خود را باز کند باید لیستی از وظایفش تهیه کند	
Postconditions:	لیست وظایف باید ثبت (saved) شود ادرس حساب کاربری باید ثبت شود لیست مشاهده شده باید با لیست از قبل ثبت شده مطابقت داشته باشد	
Flow of activities:	Actor	System
	1.1.کاربر وارد نرم افرار میشود 1.2.ادرس کاربر با ادرس حساب کاربری‌اش تطابق پیدا میکند 2.1.وارد قسمت لیست وظایف میشود 2.2.ان هارا مشاهده و در صورت نیاز ان هارا ویرایش یا تیک میزند 3.لیست نهایی را مشاهده میکند	1.1.سیستم کاربر را شناسایی میکند 1.2ادرس کاربر را با ادرس خساب های کاربری چک میکند 1.3.حساب مربوطه را نشان میدهد 2.1.لیست وظایف را به کاربر نمایش میدهد 2.2.به کاربر دسترسی ویرایش لیست و تیک گذاری را میدهد 3.1.اخرین تغییرات در لیست وظایف را ثبت میکند 3.2. لیست نهایی را نمایش میدهد
Exception events:	افتتاح حساب کاربری در صورت عدم وجود ان تهیه لیست وضایف در صورت عدم وجود ان	

Use case document no.2

Use case name:	برنامه ریزی روزانه	
Scenario:	برنامه ریزی کردن وظایف روزانه خود به طورآنلاین	
Triggering event:	کاربر وظایف روزانه خود را یادداشت می‌کند	
Brief scription:	کاربر وارد حساب کاربری می شود و به قسمت برنامه ریزی روزانه رفته و برنامه روزانه خود را یادداشت می کند	
Actors:	کاربر	
Related use case:	هدف گزاری – اولویت بندی – ویرایش برنامه روزانه	
Stakeholders:	Accounting – List – Register	
Preconditions:	کاربر قبل از برنامه ریزی وظایف روزانه خود باید وارد حساب کاربری خود بشود باید برنامه ریزی روزانه خود را با توجه به وظایفش یادداشت بکند	
Postconditions:	حساب کاربری باید ثبت شود قسمت برنامه ریزی روزانه باید در برنامه ثبت شود برنامه ریزی روزانه خود باید انجام شود	
Flow of activities:	Actor	System
	۱) کاربر یک حساب کاربری ایجاد می کند و اطلاعات اولیه خود را ثبت می کند ۲) کاربر وارد قسمت برنامه ریزی روزانه می شود و برنامه روزانه خود را با توجه به هدف و اولویت‌هایش یادداشت می کند ۳) برنامه ریزی روزانه کاربر ثبت می‌شود	۱.۱) سیستم یک حساب جدید ایجاد می‌کند ۱.۲) سیستم اطلاعات اولیه کاربر را چک می‌کند ۲.۱) سیستم قسمت برنامه ریزی روزانه را در دسترس کاربر قرار می دهد ۲.۲) سیستم امکان یادداشت کردن و اصلاح کردن جهت نوشتن وظایف روزانه خود را به کاربر می‌دهد ۳.۱) سیستم جزئیات یادداشت کاربر را چک می کند و ثبت نهایی می کند
Exception events:	۱) اطلاعات اولیه کاربر ناقص باشد و یا معتبر نباشد ۲) درخواست برنامه روزانه خود در صورت عدم ثبت آن	

Use case document no.3

Use case name:	ویرایش وظیفه.	
Scenario:	دانشجو پس از بازبینی وظایف خود اقدام به ویرایش انها میکند.	
Triggering event:	دانشجو وقتی وظایف روز خود رامرور میکند متوجه میشود که وظایف خود نیاز به تغییراتی دارد.	
Brief scription:	دانشجوپس از ورود به پنل کاربری خود در برنامه time tact وظایف از پیش تعیین شده ی خود را تغییر میدهد و ان را ثبت نهایی میکند.	
Actors:	دانشجو.	
Related use case:	اولویت بندی وظایف - مدیریت وظایف - مشاهده لیست وظایف	
Stakeholders:	دانشجو – مدیر بانک مرکزی – ادمین برنامه – پشتیبان .	
Preconditions:	دانشجو باید حساب کاربری داشته باشد . دانشجو باید لیست وظایفش را ثبت کند	
Postconditions:	باید در موعد های تعیین شده اقدام به تکمیل وظایف کند .دانشجو باید توافق نامه مربوط به تشویق و تنبیه مربوط به تکمیل وظایف خود را قبول کند . دانشجو باید پس از اتمام هر وظیفه ان را ثبت کند . دانشجو باید فقط و فقط خود به پنل کاربری وارد و نسبت به انجام وظایف خود اقدام کند .	
Flow of activities:	Actor	System
	1.دانشجو باید حساب کاربری فعال داشته باشد . 2.دانشجو باید شب قبل یا نهایتا صبح وظایف روز خود را وارد کند . 3.دانشجو باید وظایف خود را با جزییات وارد کند. 4.دانشجو باید مشخصات حساب بانکی را وارد کند.	1.سیستم باید تطابق اطلاعات وارد شده با اطلاعات پیشین دانشجو را انجام دهد. 2.سیستمباید رمز یک بار مصرف را در هر بار ورود دانشجو به حساب خود را به شماره موبایل او ارسال کند. 3.سیستم باید پس از هر بار ویرایش اطلاعات توسط دانشجو پیام تاییدیه ای برای شماره موبایل او ارسال کند . 4.سیستم باید وظایف وارد شده را بازبینی و در صورت خطا و تداخل زمانی ای پیام اخطار را به دانشجو نشان دهد .
Exception events:	دانشجو قادر به مشاهده لیست وظایف خود نباشد دانشجو قادر به ویرایش لیست وظایف خود نباشد	

Use case document no.4

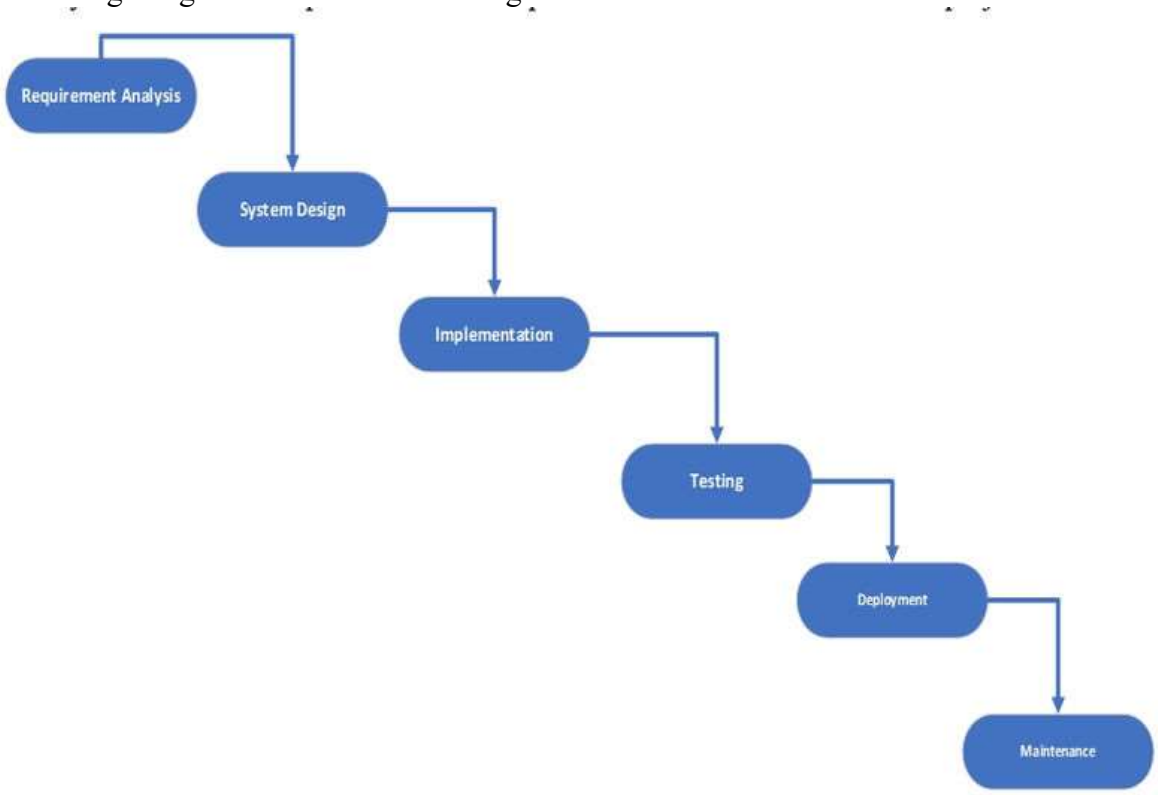
Use case name:	یاد آوری وظایف	
Scenario:	فعال کردن اعلان به طور آنلاین برای یادآوری	
Triggering event:	کاربر عنوان وظایف خود را یادداشت می کند برای یاد اوری	
Brief scription:	کاربر به صفحه کاربری خود رفته و وارد قسمت یادآوری وظایف می شود و اعلان وظایفی که یادداشت کرده را فعال می کند تا به آن یادآوری کند	
Actors:	کاربر	
Related use case:	فعال کردن اعلان تایم بندی کردن وظایف ویرایش اعلان یادآوری وظایف	
Stakeholders:	کاربر _لیست_زنگ اعلان	
Preconditions:	کاربر برای فعال کردن یادآوری وظایف خود باید از قبل حساب کاربری خود را باز کند و عنوان اعلام خود را مشخص کند و آن ها را فعال کند جهت یاد اوری	
Postconditions:	لیست یاد اوری وظایف باید ثبت شود بعد عنوان وظایف جهت یادآوری ثبت شود فعال کردن اعلان یادآوری وظایف	
Flow of activities:	Actor	System
	۱_کاربر جهت ایجاد حساب کاربری باید اطلاعات اولیه خود را ثبت کند ۲_کاربر با توجه به هدف و اولویت خود اعلان و یادآوری وظایف خود را ثبت می کند ۳_اعلان یادآوری وظایف کاربر ثبت می شود	۱_۱سیستم اطلاعات اولیه کاربر را شناسایی می‌کند ۲-۱سیستم حساب جدید کاربری را ثبت می کند ۲_۱سیستم لیست یادآوریوظایف رو در دسترس کاربرقرار می دهد ۲_۲به کاربر فعال کردن اعلان خود را می دهد ۳_۱سیستم جزئیات اعلان وظایف کاربر زا چک می‌کند ۳_۲سیستم لیست نهایی یادآوری وظایف را ثبت می‌کند
Exception events:	در صورت مشخص نکردن عنوان یادآوری وظایف نمی توان اعلان را فعال کرد درخواست یادآوری وظایف در صورت عدم ثبت آن	

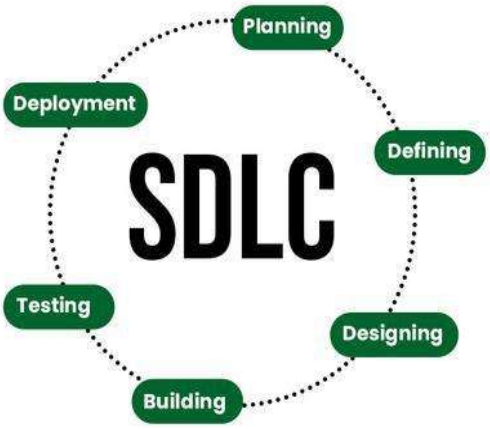
PROJECT EXECUTION METHOD

Waterfall model:

The classic waterfall model is the main software development life cycle model. It is very simple but idealistic. It was very popular in the past but is not used today. However, it is very important because all other software development life cycle models are based on the classic waterfall model. It is very easy to understand and use. The waterfall model shows the software development process in a linear sequential flow. Hence, in the waterfall model, each phase must be completed before the next phase starts and there is no interference in the phases and the phases do not overlap in any way.
The waterfall model is a software development model that is used in the context of simple and light projects with uncertainty. It is characterized by a structured and sequential approach to project management and software development. The waterfall model is useful in situations where the project requirements are well defined and the project objectives are clear.
The Waterfall approach was the first SDLC model that was widely used in software engineering to ensure the success of the project.

The following image shows the different stages of the waterfall model:





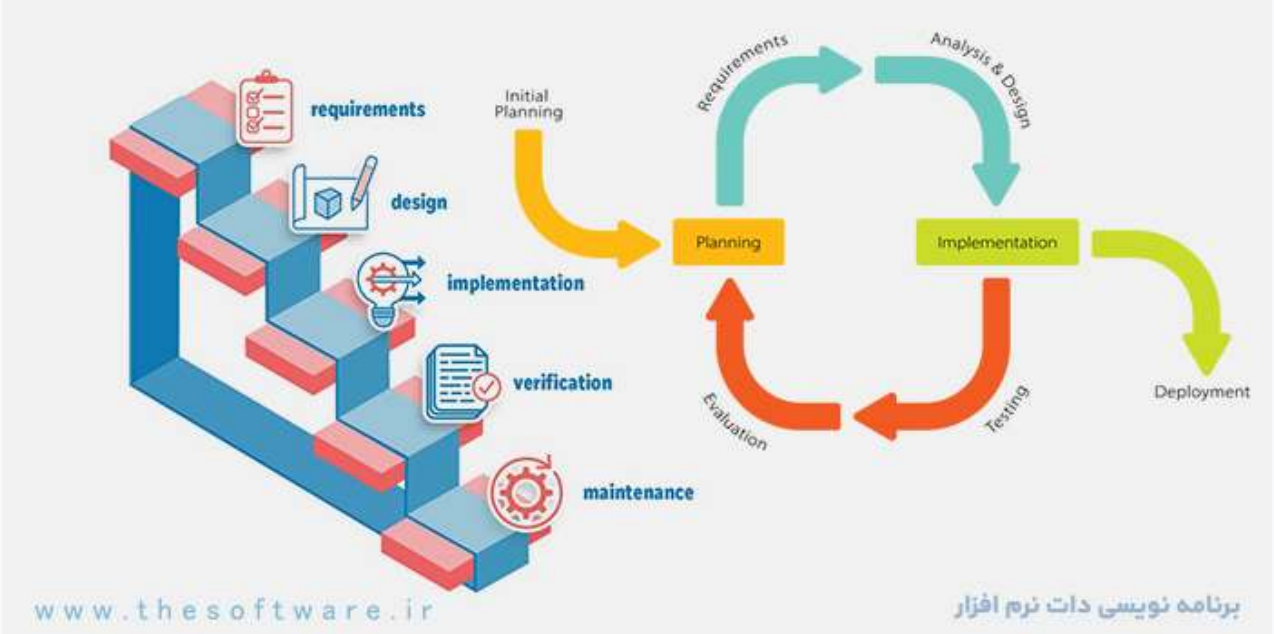
The Software Development Life Cycle (SDLC) is a structured process used to design, develop, and test good quality software. SDLC, or Software Development Life Cycle, is a methodology that defines the entire software development process step by step.

Our goal in choosing this method for developing our software was to implement a simple and small student project, which was perfectly suited to the features mentioned above. Also, since we exactly carried out the project phase by phase and proceeded step by step, it is exactly consistent with the waterfall model, so this reason was also a good option for choosing it.
But since we do not intend to enter the construction and coding phase of the project, it gives us the power to choose between the waterfall method and the Agile model, because the Agile method is more suitable for large and buildable projects. Our project was simple and small with specific requirements and step by step.
In project implementation, progress flows continuously downward (like a waterfall) among the phases. The next phase begins only after achieving the set of goals defined for the previous phase and signing it, just like the way we carried out the project and tasks step by step and moved forward step by step without overlapping.

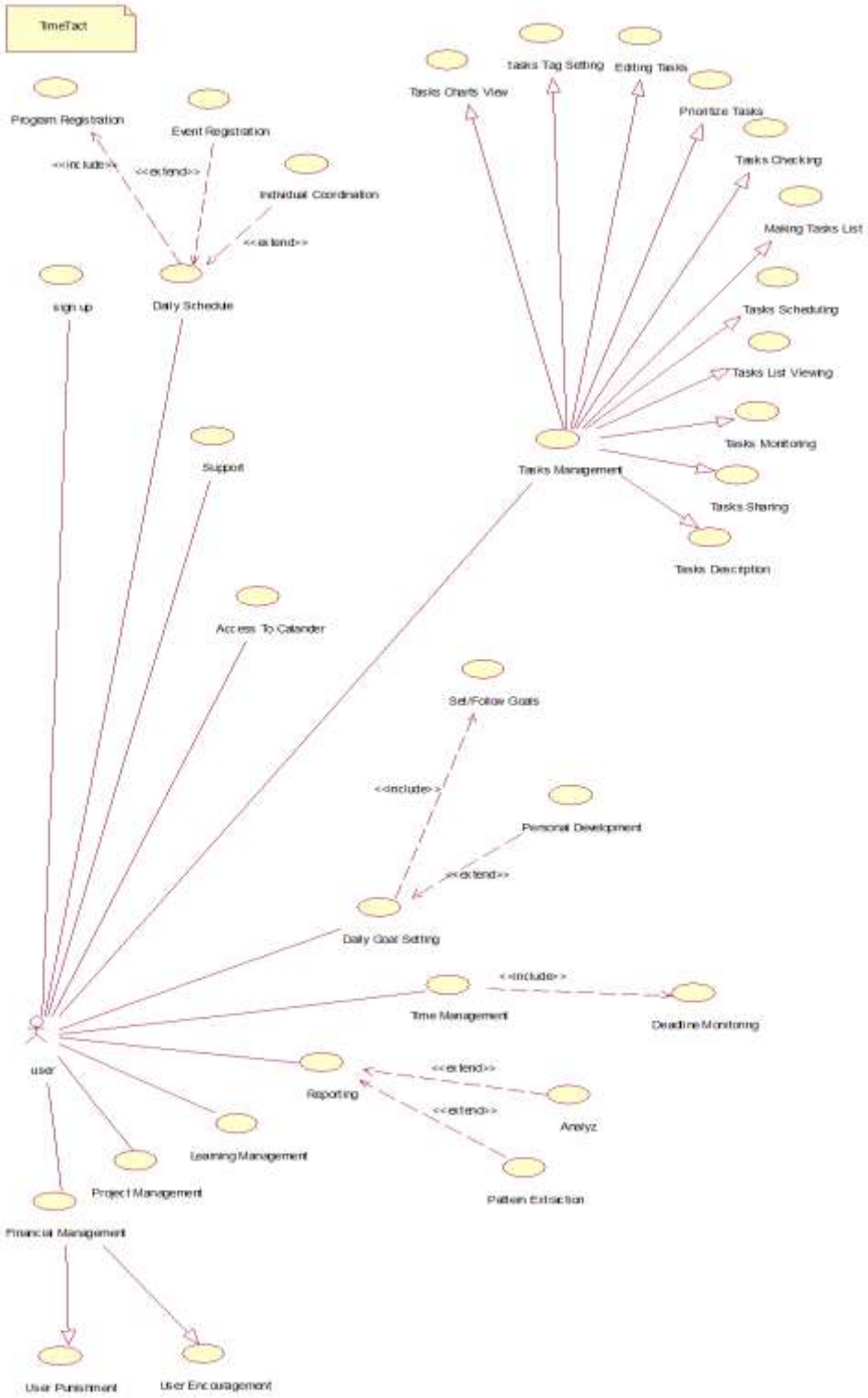
Other features for choosing the waterfall model for TimeTact software development:

Requirements are very well documented, clear and fixed and not ambiguous at all.
The product definition is fixed.
The technology is understandable and not dynamic.
Sufficient resources with the required expertise are available to support the product.
The project is short, small and lightweight.
The simplicity and comprehensibility of the model for everyone.
It allows for segmentation and control and gives the project managers the opportunity to fully troubleshoot the previous stage before starting the next one.
Easy to manage due to the robustness of the model.
Each stage has specific deliverables and a review process.
The stages are clearly defined.
Easy to organize tasks.

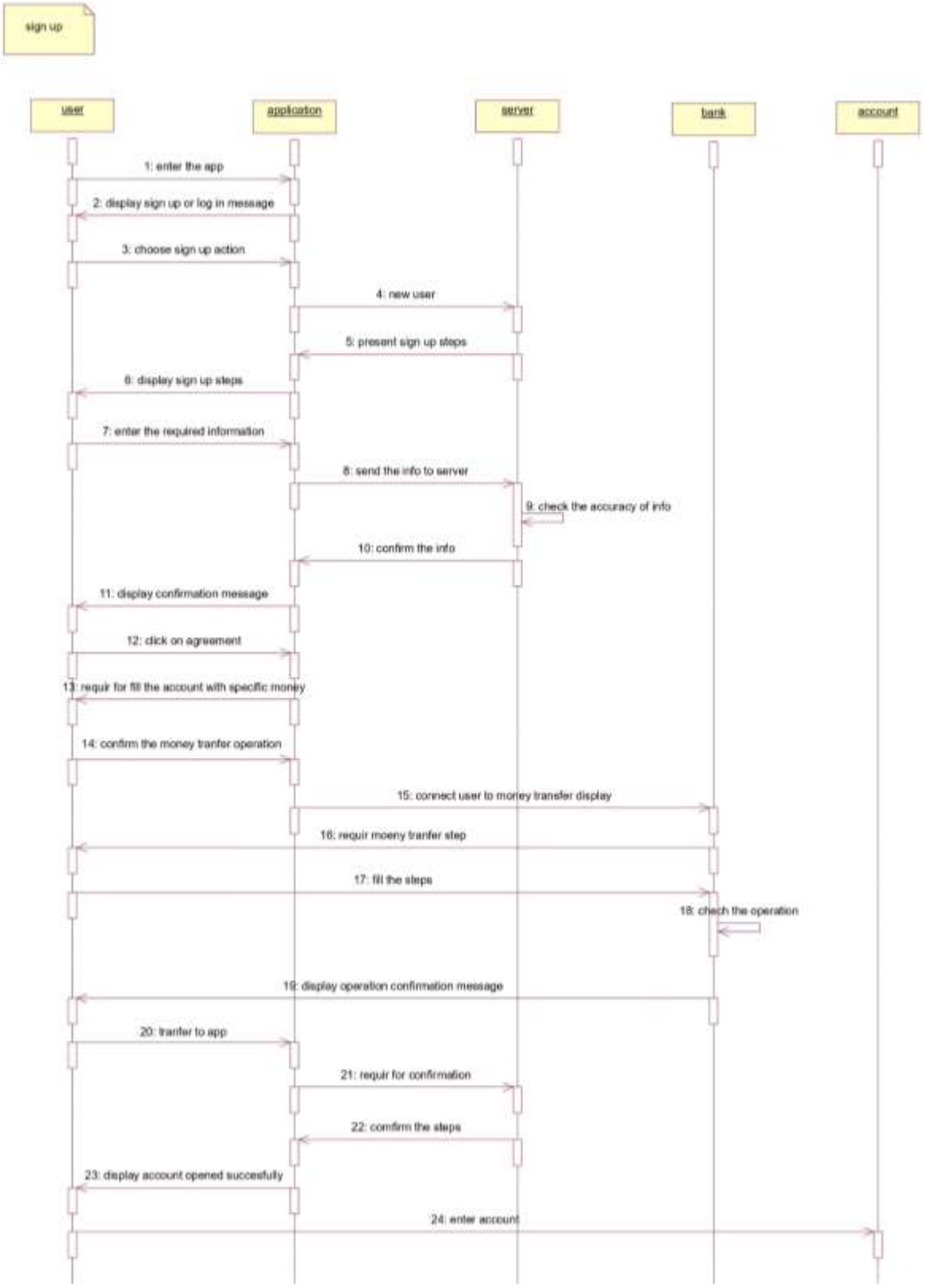
We used these features to implement the Aman project and for these reasons we prioritized the waterfall model over other models for selection.



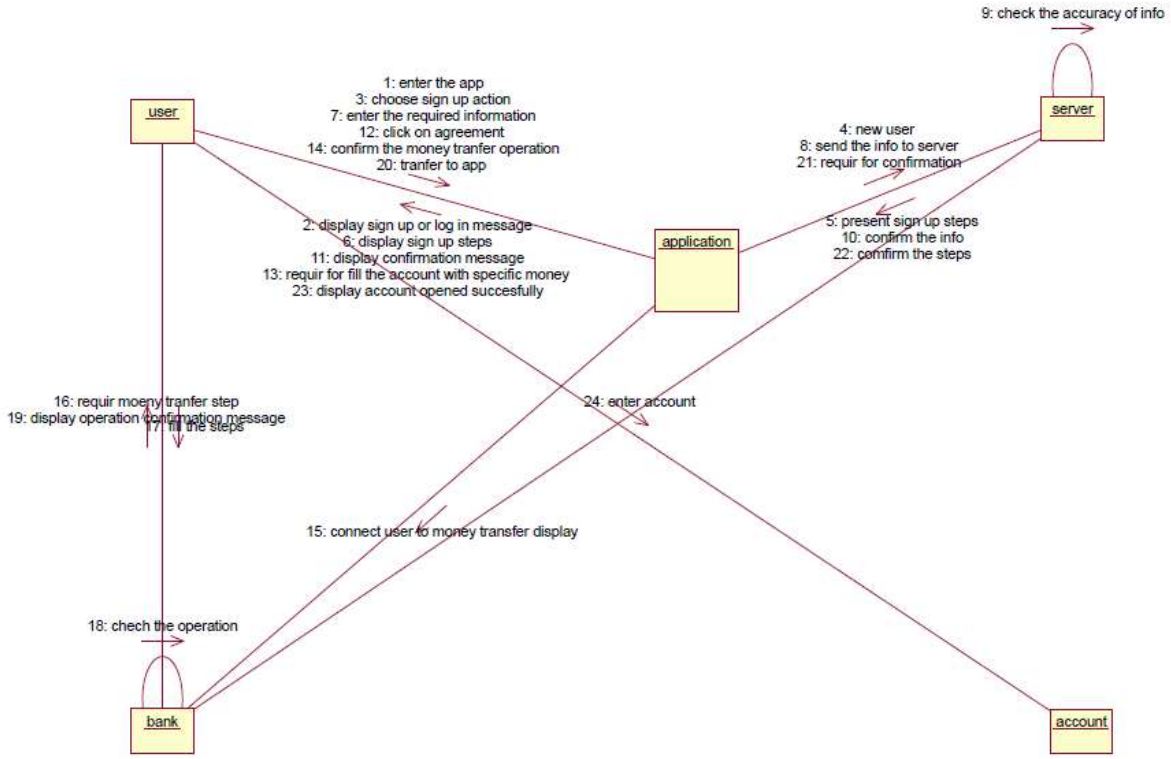
Use case diagram



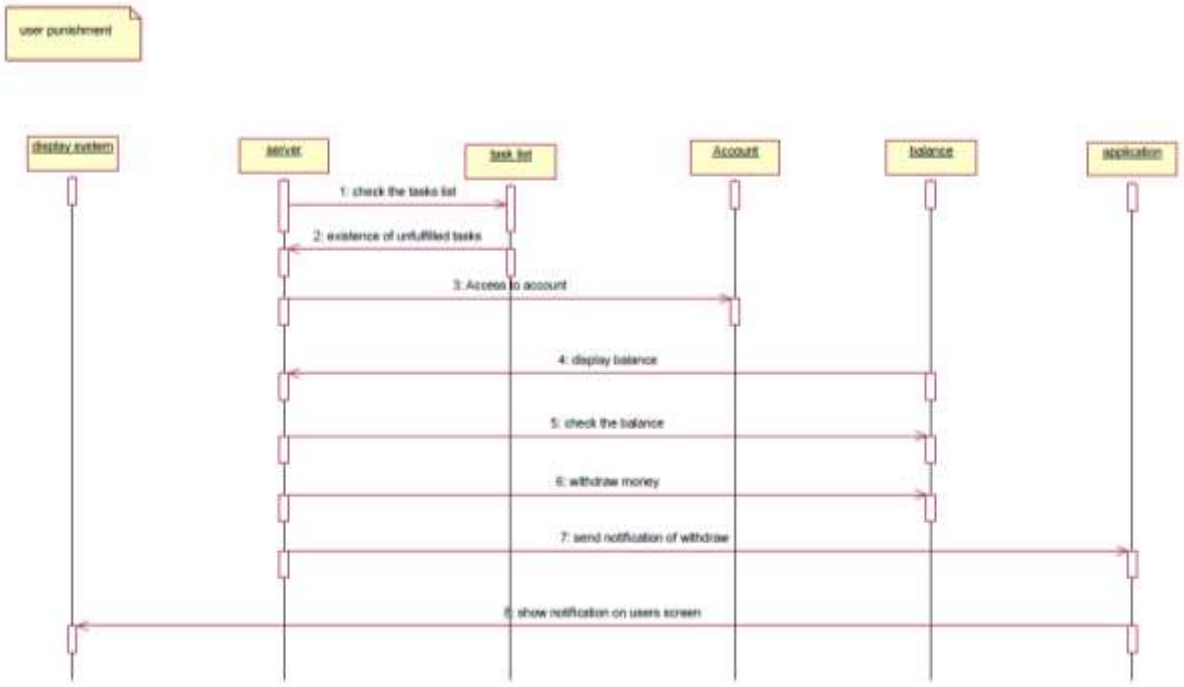
Sequence diagram 1 – sign up



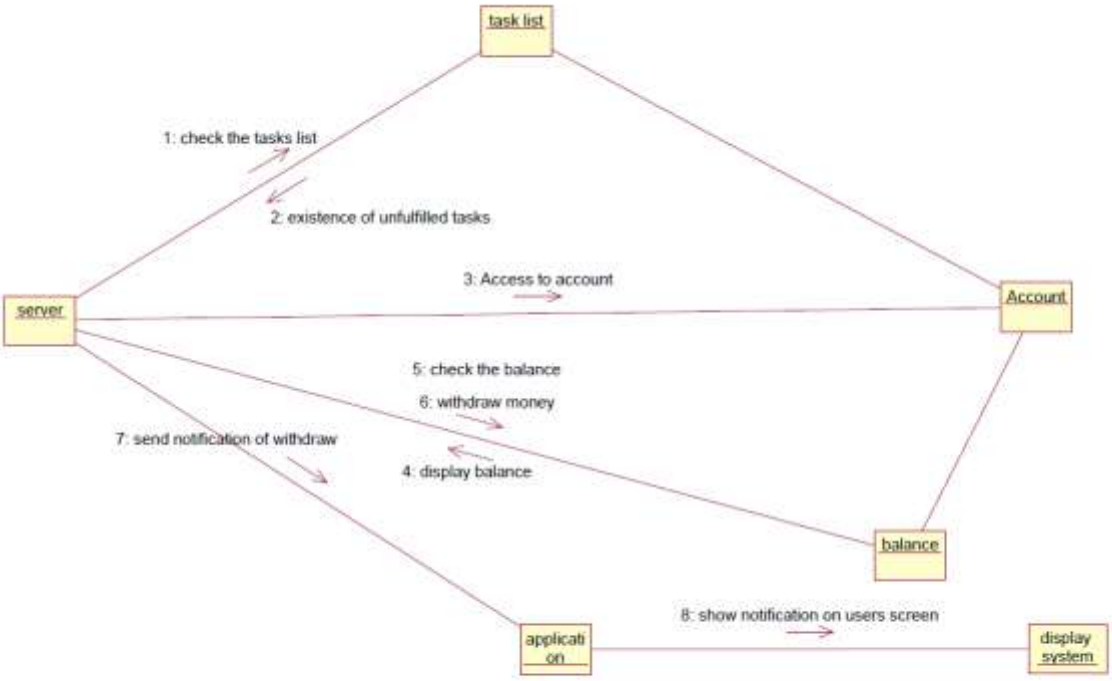
Collaboration diagram 1 – sign up



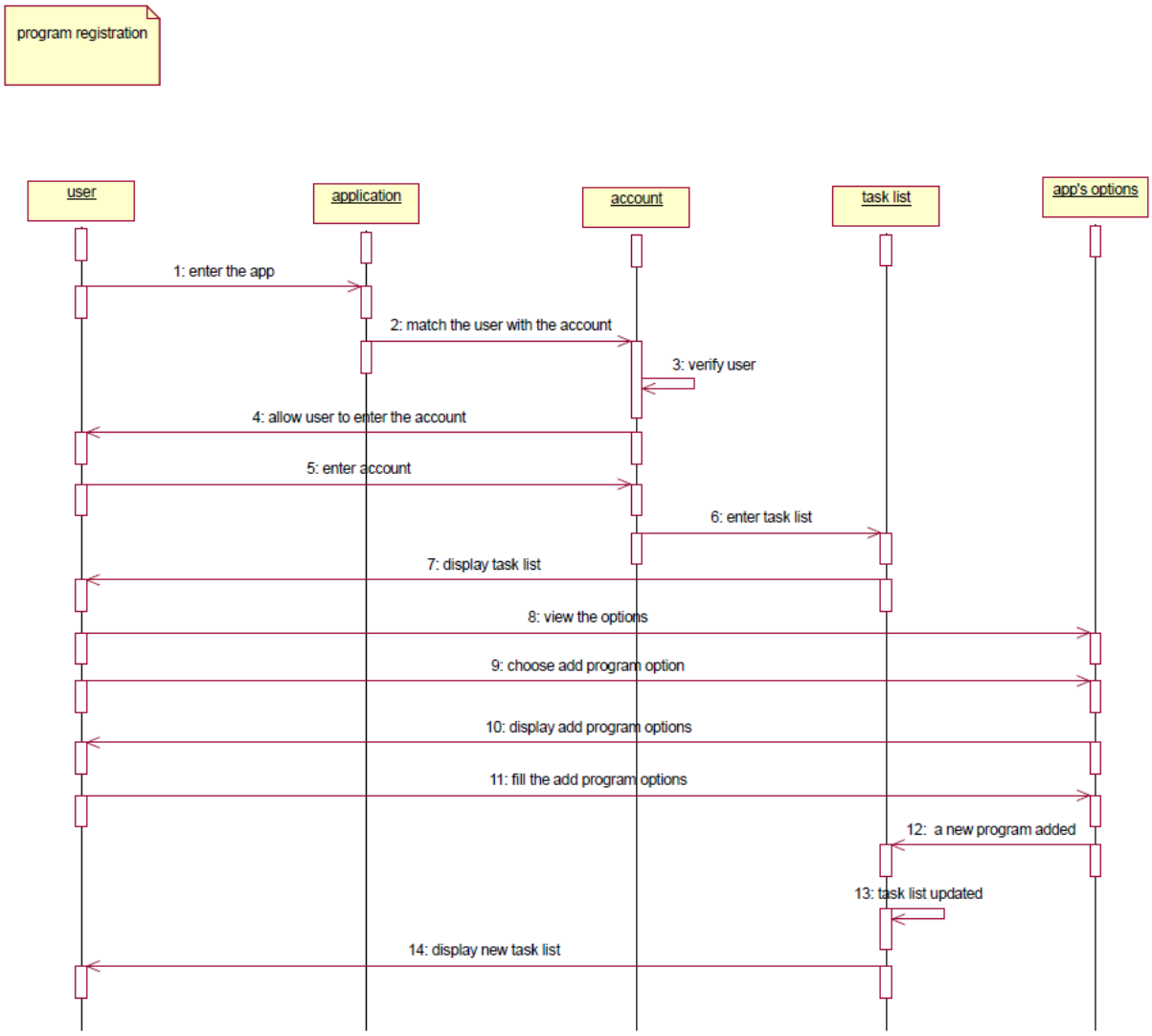
Sequence diagram 2 – user punishment



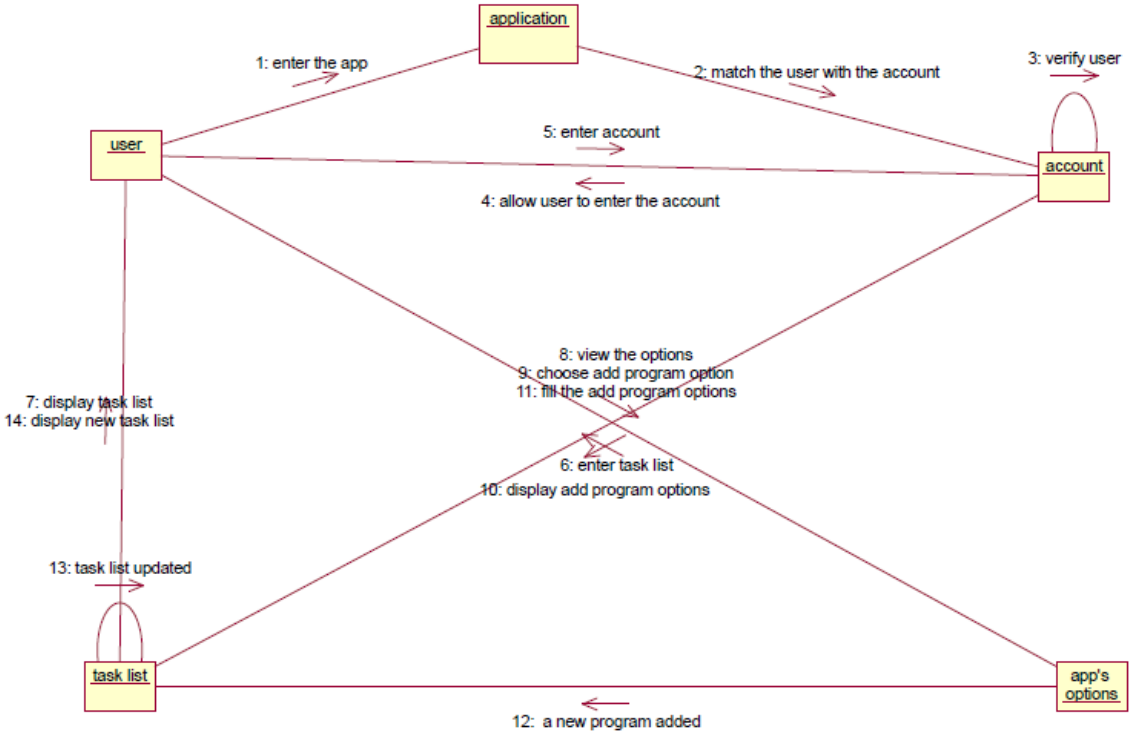
Collaboration diagram 2 – user punishment



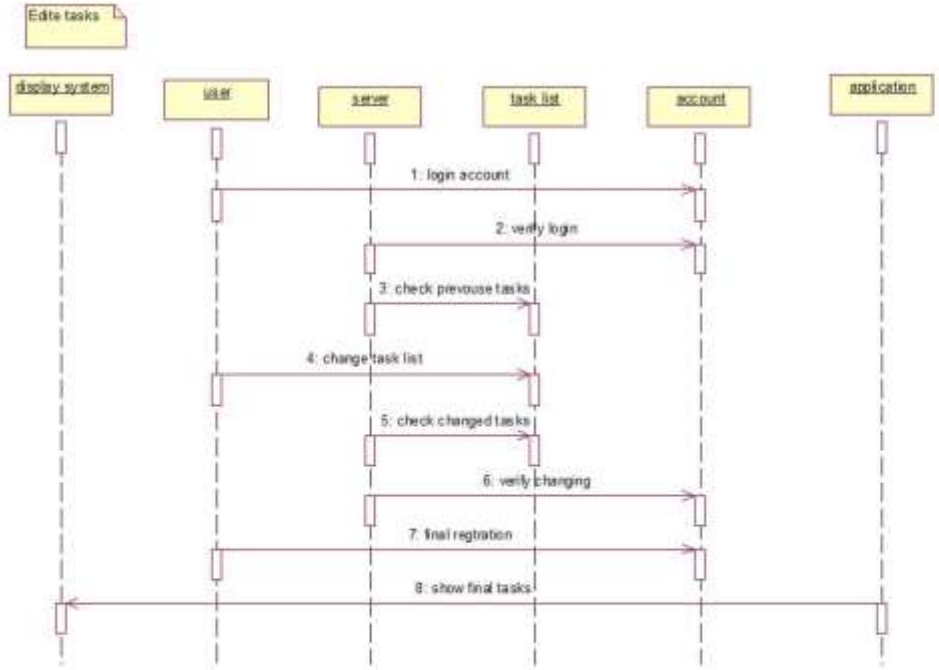
Sequence diagram 3 – program registration



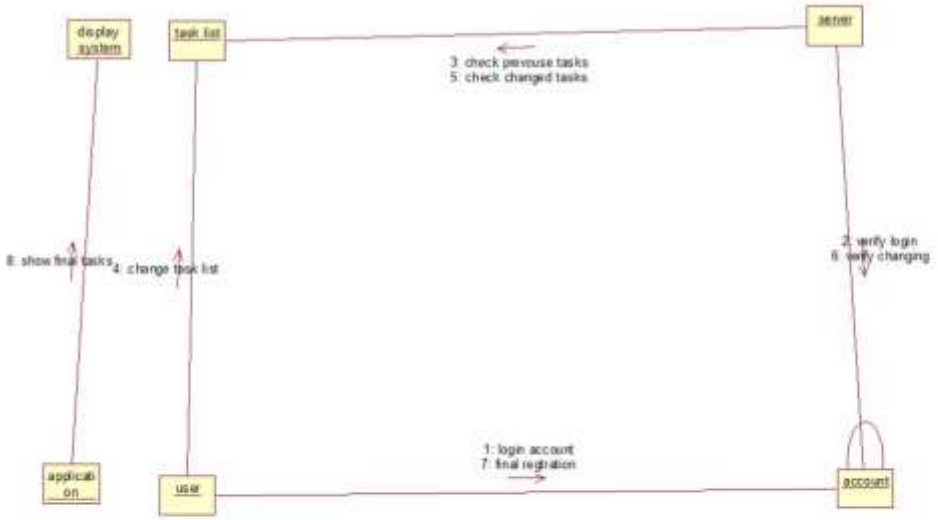
Collaboration diagram 3 – program registration



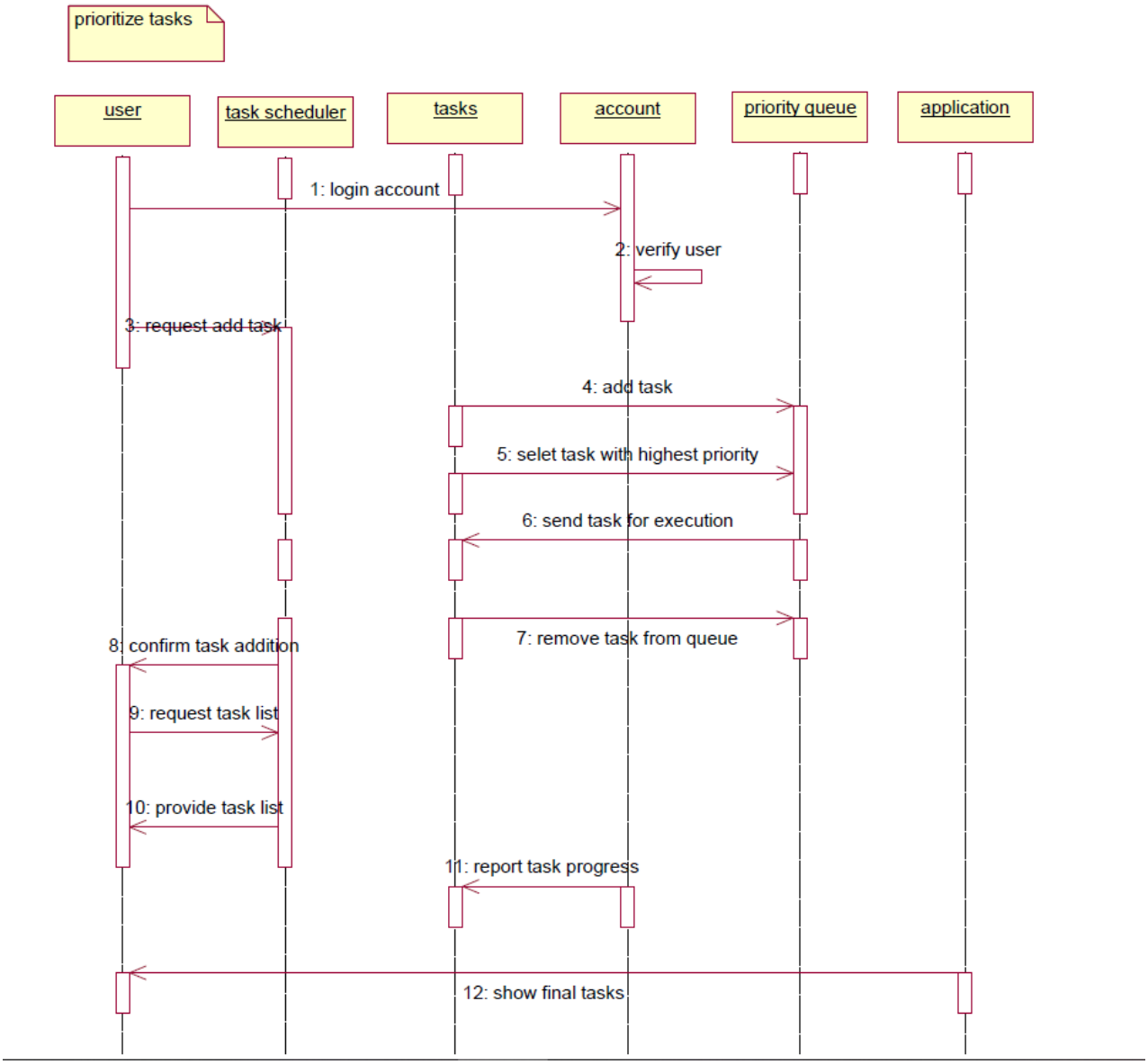
Sequence diagram 4 – edit tasks



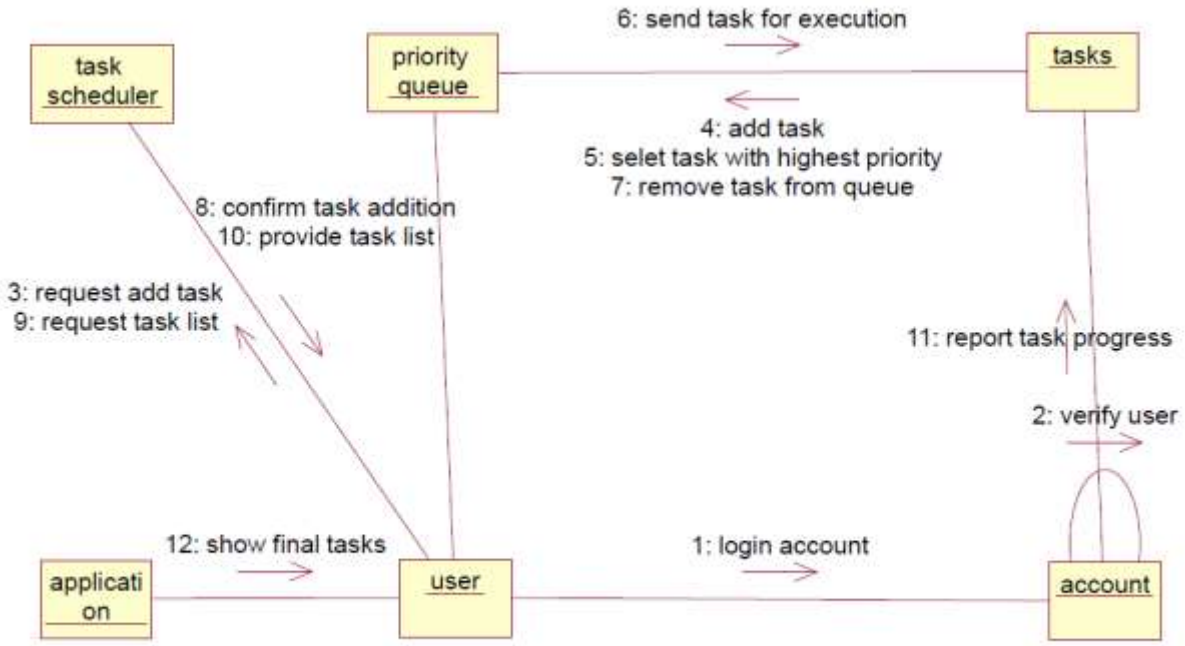
Collaboration diagram 4 – edit tasks



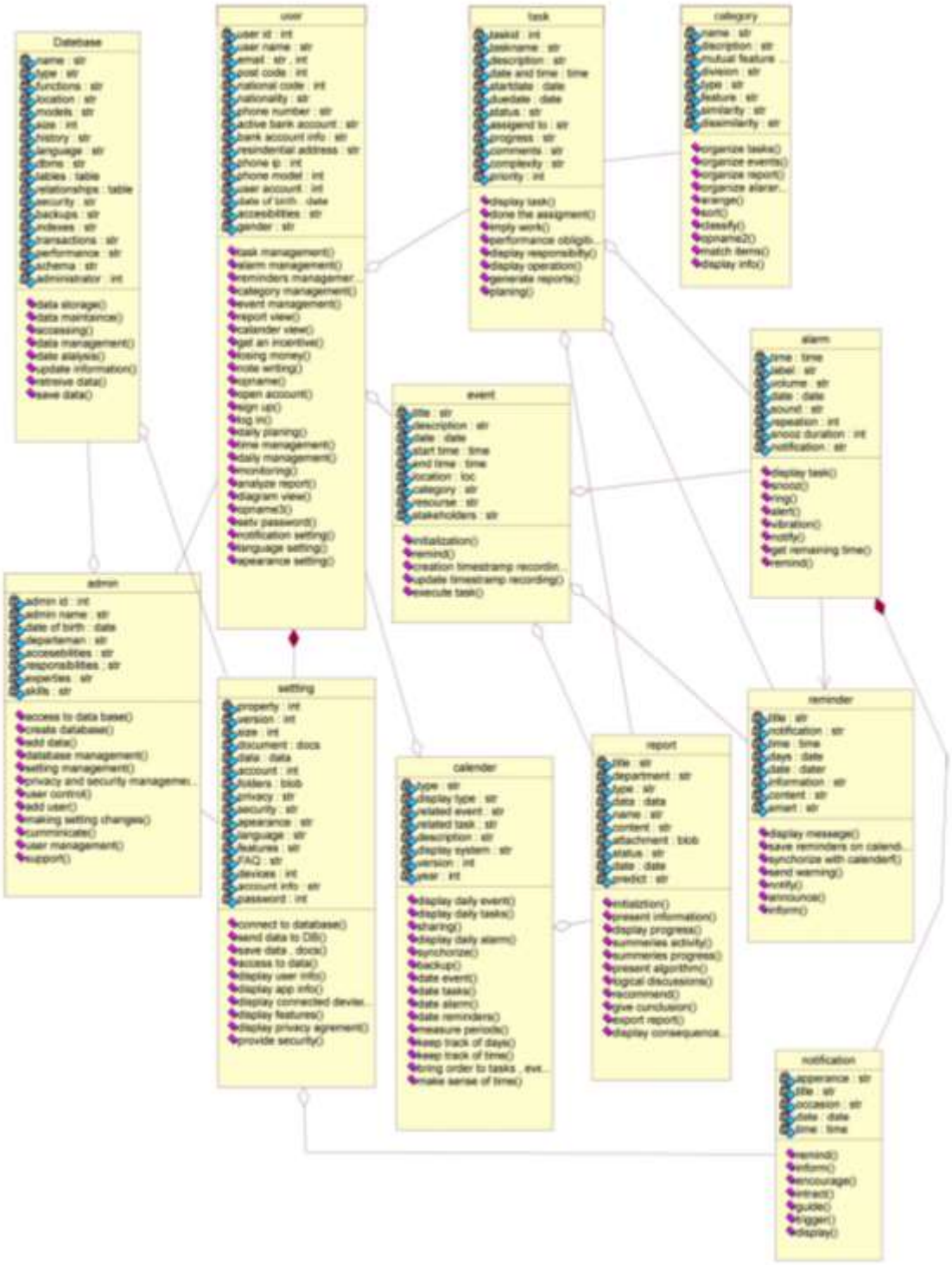
Sequence diagram 5 – prioritize tasks



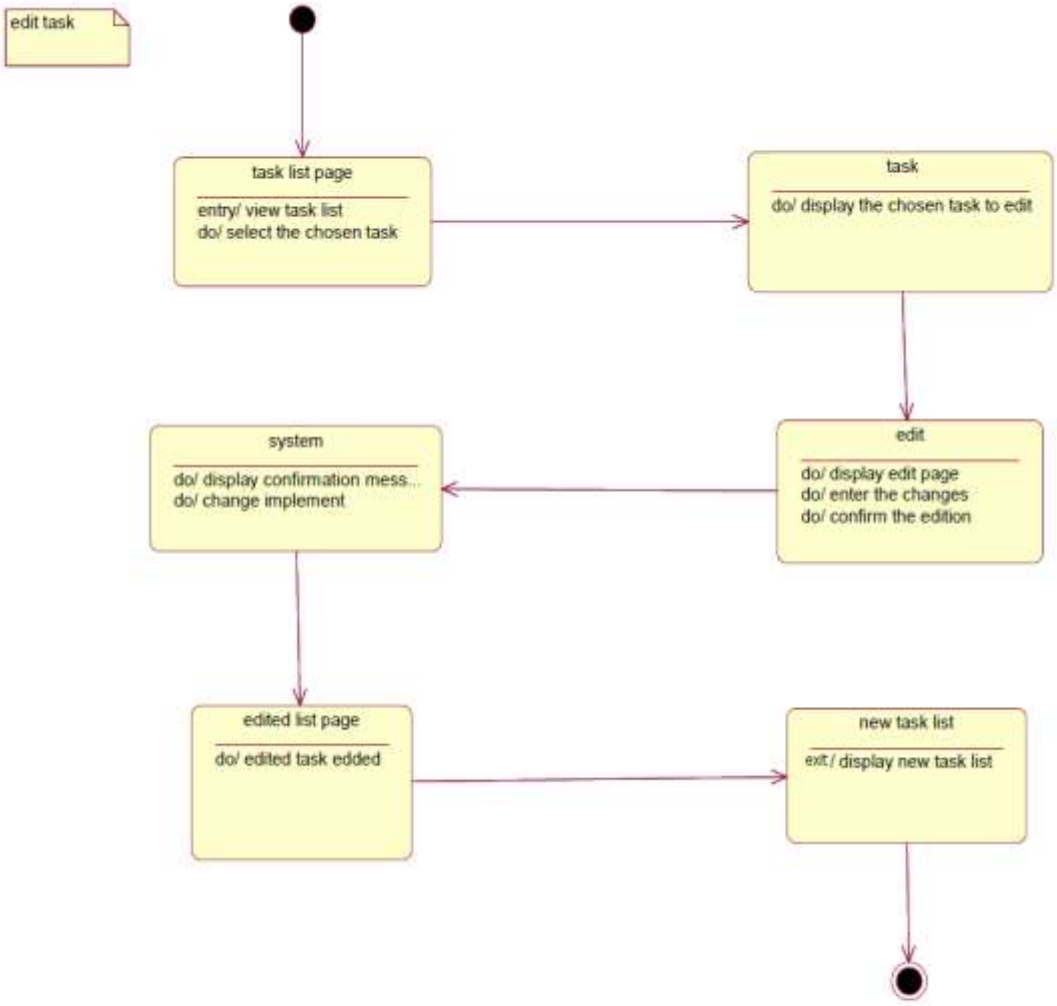
Collaboration diagram 5 - prioritize tasks



Class Diagram



State Chart Diagram – edit tasks



Activity Diagram – edit tasks