



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ

ФАКУЛТЕТ КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ

**СПЕЦИАЛНОСТ: КОМПЮТЪРНО И СОФТУЕРНО
ИНЖЕНЕРСТВО**

КУРСОВА РАБОТА

Дисциплина: UML ООП

Тема: “Система за планиране и организиране на пътуване”

Образователно-квалификационна степен: Магистър

Студент:

Божидар Николаев Захов /...../

фак. №: 121323044, група: 222

e-mail: bzahov1998@gmail.com

LinkedIn: <https://www.linkedin.com/in/bzahov98/>

хн. ас.: инж. Зеки Малунски /...../

София, 2024

1. Съдържание

| | |
|--|-----------|
| 1. Съдържание..... | 2 |
| 2. Въведение..... | 3 |
| 3. Бизнес спецификации..... | 3 |
| 4. Технически спецификации..... | 4 |
| 4.1. Kotlin Multiplatform:..... | 4 |
| 4.2. Backend (Spring Boot):..... | 4 |
| 4.3. Frontend (Android, iOS, уеб):..... | 4 |
| 5. Use case диаграми..... | 5 |
| 5.1. Неоторизиран потребител..... | 5 |
| 5.2. Регистриран потребител (турист)..... | 6 |
| 5.3. Регистриран потребител (партньор)..... | 7 |
| 6. Class диаграми..... | 8 |
| 7. Sequence диаграми..... | 9 |
| 7.1. Вписване на потребителя..... | 9 |
| 7.2. Планиране на пътуване..... | 10 |
| 7.3. Разглеждане и обновяване на обект..... | 11 |
| 8. Communication диаграми..... | 12 |
| 8.1. Посетител..... | 12 |
| 8.2. Партньор..... | 12 |
| 9. StateMachine..... | 13 |
| 9.1. Вписване..... | 13 |
| 9.2. Създаване на Партньорски Обект..... | 13 |
| 9.3. Създаване на ново пътуване като Турист..... | 13 |
| 10. Activity диаграми..... | 14 |
| 10.1. Вписване..... | 14 |
| 10.2. Промяна на пътуването на турист..... | 15 |
| 10.3. Добавяне на оценка на партньорски обект..... | 16 |
| 11. Deployment + Package диаграма..... | 17 |
| 12. Използвана литература:..... | 17 |

2. Въведение

В документа се описват характеристиките на **система за планиране и организиране на пътувания**. Тя представлява онлайн платформа, в която потребителите могат да се регистрират, както и да планират, организират и да получават съвети и предложения за бъдещите си пътувания, които могат да са разпръснати по различни градове, а партньорите на платформата - да рекламират своите обекти.

3. Бизнес спецификации

В системата има два вида типа потребители:

- **Редовен потребител - турист:**

В конкретните градове, потребителите могат да организират по часове, своето време и така да посетят възможно най-много туристически обекти, атракции и междувременно да се отбият в партньорски на платформата обекти, които да бъдат рекламирани и оценявани от клиентите.

- **Администратор на обекти - партньор на платформата**

В платформата има възможност **партньори да рекламират своите обекти**, които клиентите могат също да добавят, в своите планове, от което платформата да печели.

Системата предоставя **интеграция** с най-популярните **социални мрежи**, с цел по-голямо **популяризиране на туристическите атракции и комерсиалните обекти** и по-лесен начин за **споделяне** между хората за различните **маршрути** и примерни **планове**, включващи реда и необходимото време за посещение на отделните обекти.

Например един потребител е посетил даден град, направил е примерен план за пътуването и го е разделил на 3 дни, като е описал в кой ден, какво да се посети.

Платформата също позволява **добавяне на оценки**, под формата на коментари, от страна на потребителите, във връзка с различните обекти.

Представете си това приложение, като **безценен помощник за организиране на Вашето пътуване**, който с времето започва да **опознава Вашите предпочитани обекти** и да Ви предлага **незабравимо и добре планирано пътуване**, без да се чудите:

“Ами сегаа, накъде??”.

Това ще **спести** и **оползотвори** **ценното** Ви **време** и ще се чувствате удовлетворени от новите спомени!

4. Технически спецификации

4.1. Kotlin Multiplatform:

- Използване на **Kotlin Multiplatform** за **споделяне** на **бизнес логика и модели** между мобилните приложения за Android, iOS и уеб платформата.
- **Споделен код:** Написвайки основната логика на вашето приложение или библиотека на Kotlin, то тя може да бъде използвана на различни платформи. Този код обикновено е платформено независим и съдържа бизнес логиката, алгоритмите и моделите на данни, които без проблем могат да се преизползват.[1]
- **Целта на Kotlin Multiplatform не е да премахне изцяло специфичния за различните платформи код.** Това е така, тъй като при някои платформи, има специфични изисквания, които налагат създаването на специфичен код за тях.[1]
- Вместо това той предоставя начин за максимално увеличаване на преизползването на кода, като същевременно позволява гъвкавост за специфични за платформата оптимизации и персонализации.[1]

4.2. Backend (Spring Boot):

- Използване на **Spring Boot framework** и езика **Kotlin** за създаване на **RESTful API**, което ще бъде достъпно от **frontend-a**(Android, iOS, уеб).
- **Интеграция с база данни** за съхранение на потребителска информация, пътувания и обекти.
- Използване на **JWT token** за автентикация и оторизация, осигурявайки сигурност на API - то.
- Възможност за **гъвкавост, мащабиране и оптимизиране** на управлението на трафика към бекенда, чрез API-то, използвайки вградените инструменти на **Spring Boot**.

4.3. Frontend (Android, iOS, уеб):

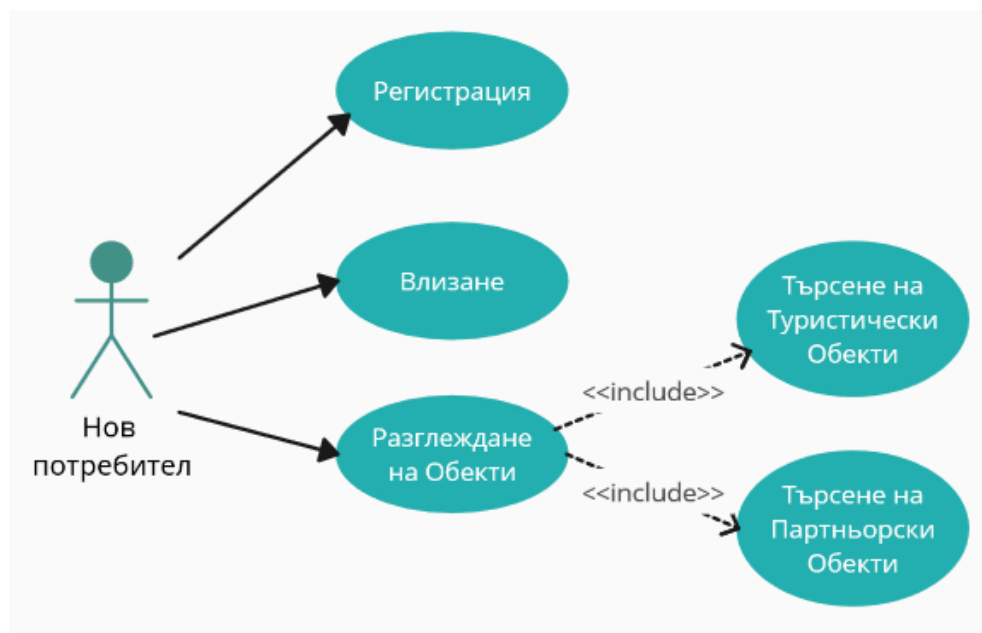
- Създаване на **Android приложение** чрез **Kotlin Multiplatform**, позволяващо лесна интеграция с устройствените функции на Android, чрез **Compose Multiplatform**.
- Разработване на **iOS приложение** чрез **Kotlin Multiplatform**, което гарантира сходен потребителски интерфейс и функционалност с Android версията.
- Изграждане на **уеб приложение** с помощта на **Kotlin Multiplatform**, което осигурява съвместимост с различните браузъри и устройства.

Тези технически характеристики гарантират ефективното функциониране на системата за планиране и организиране на пътуване, като същевременно предлагат уеднаквена функционалност и потребителски интерфейс, за всички целеви платформил.

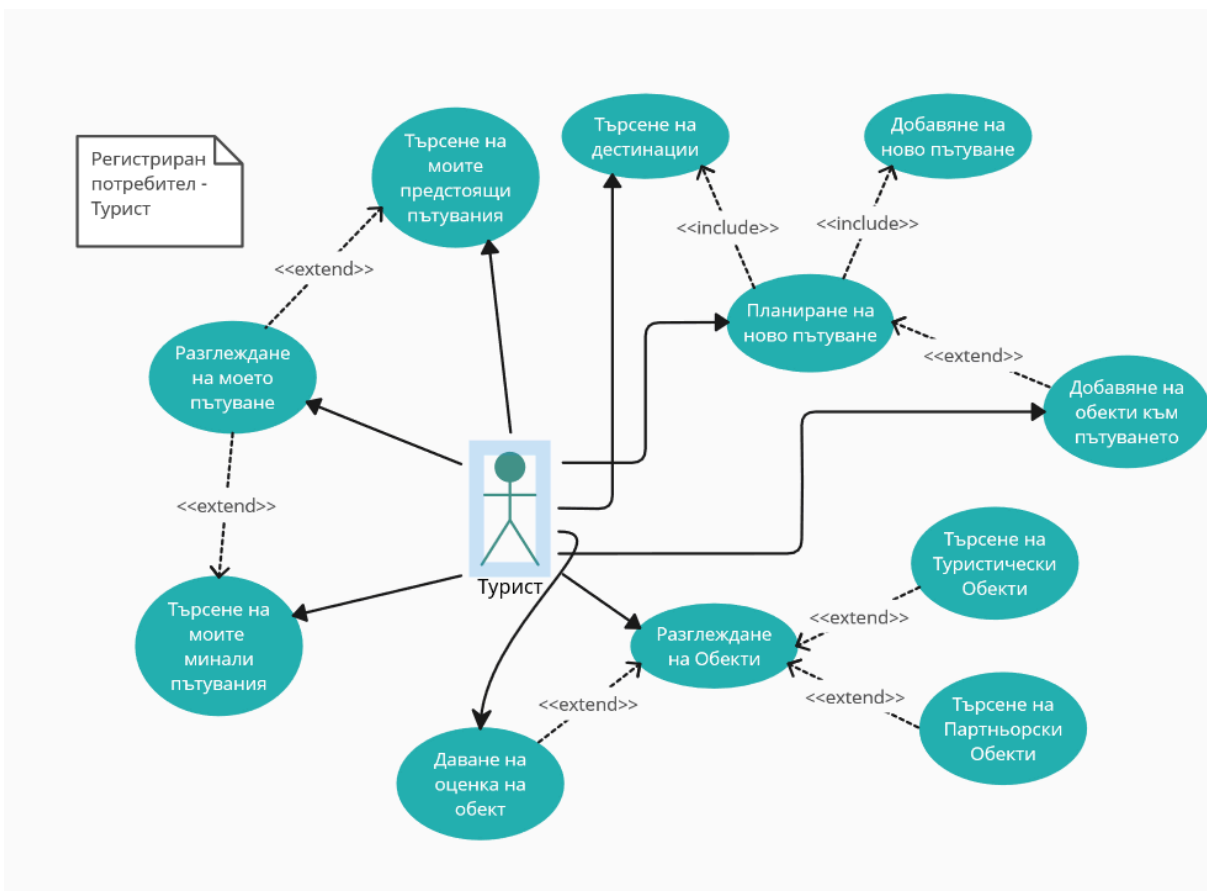
Така се намалява до минимум дублиращия се код, ускорявайки етапа на разработка и бързото вкарване в експлоатация на продукта.

5. Use case диаграми

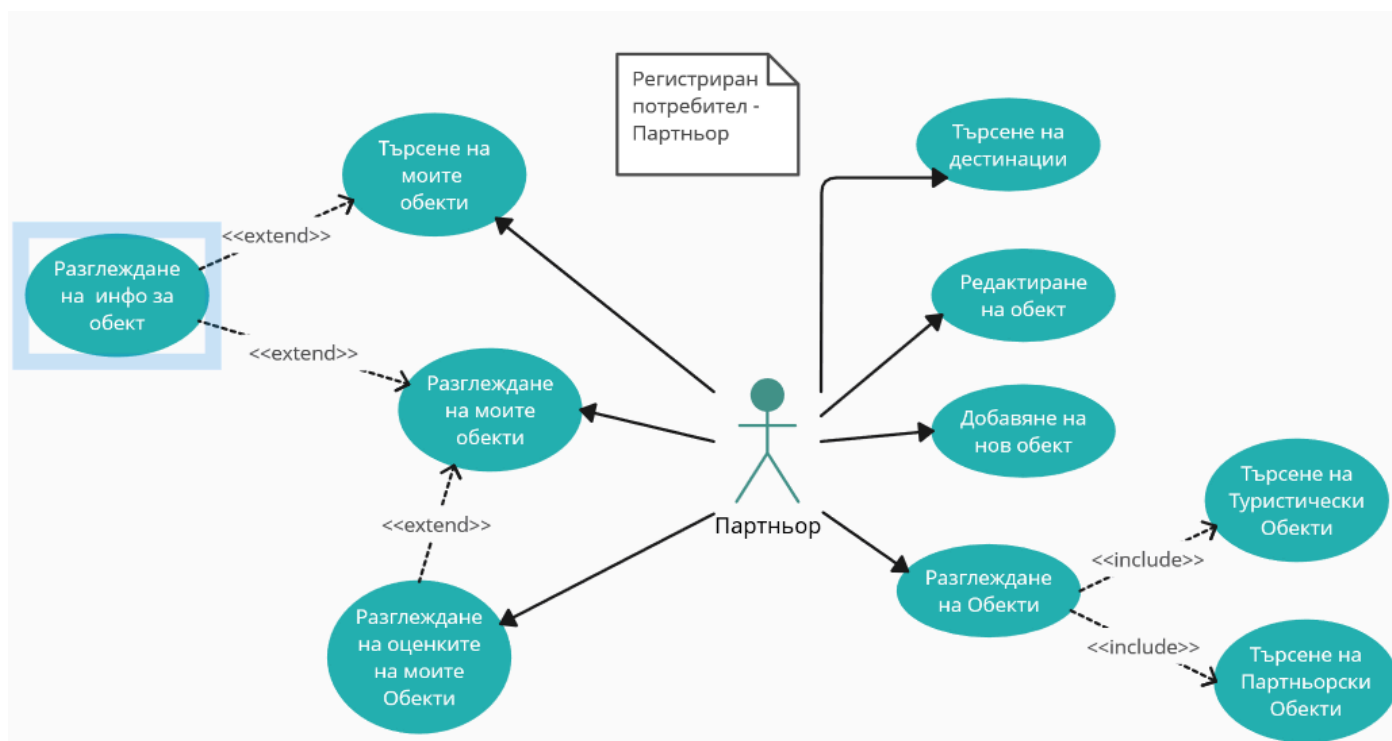
5.1. Неоторизиран потребител



5.2. Регистриран потребител (турист)



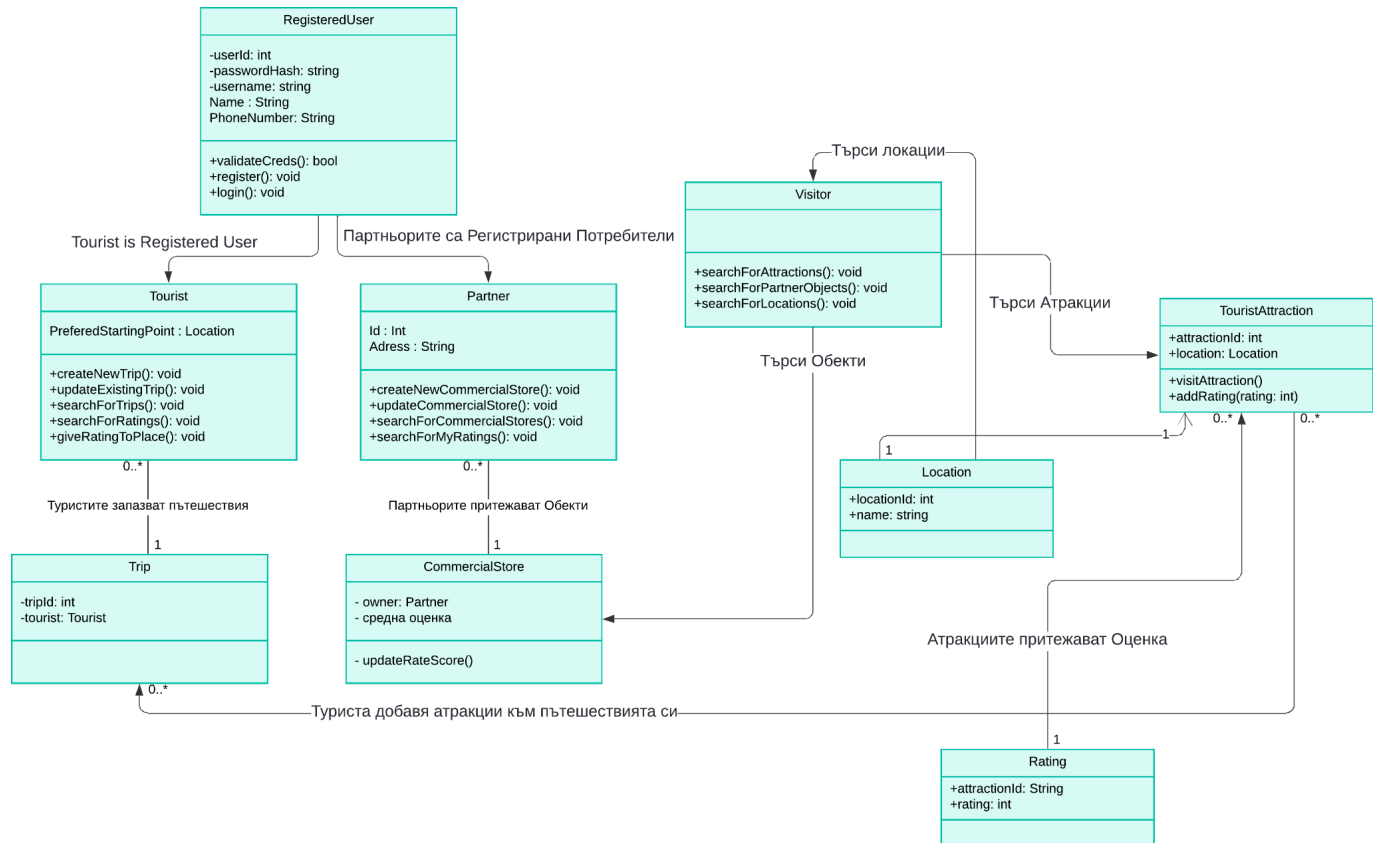
5.3. Регистриран потребител (партньор)



6. Class диаграми

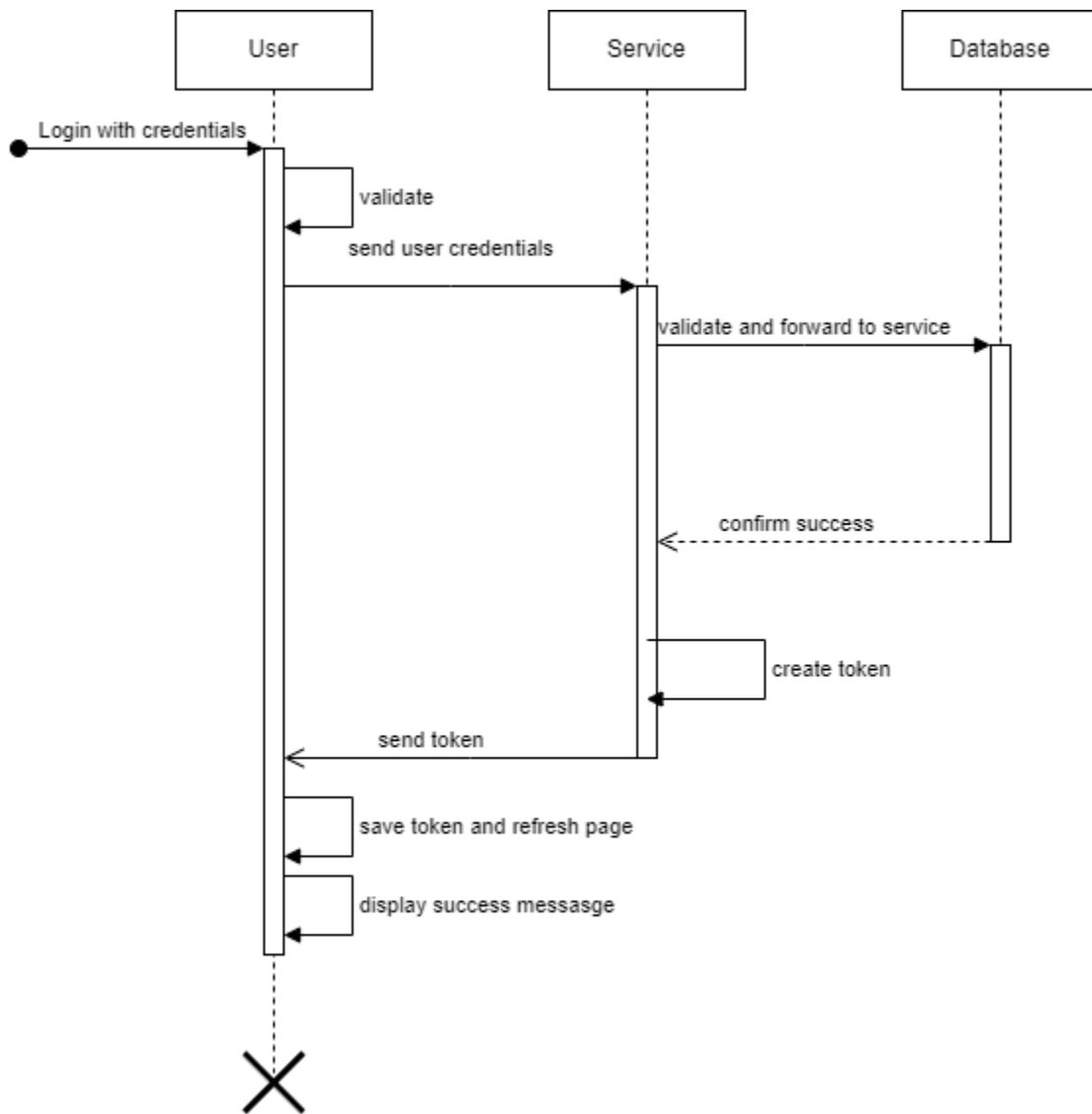
Класовете са: Visitor, RegisteredUser, Tourist, Partner, Attraction, CommercialStore, Rating, Trip,

Имаме три основни типа потребители: туристи, които създават и управляват свои пътувания, партньори, които създават и управляват търговски обекти, и посетители, които могат да търсят атракции и обекти, но не могат да ги създават.



7. Sequence диаграми

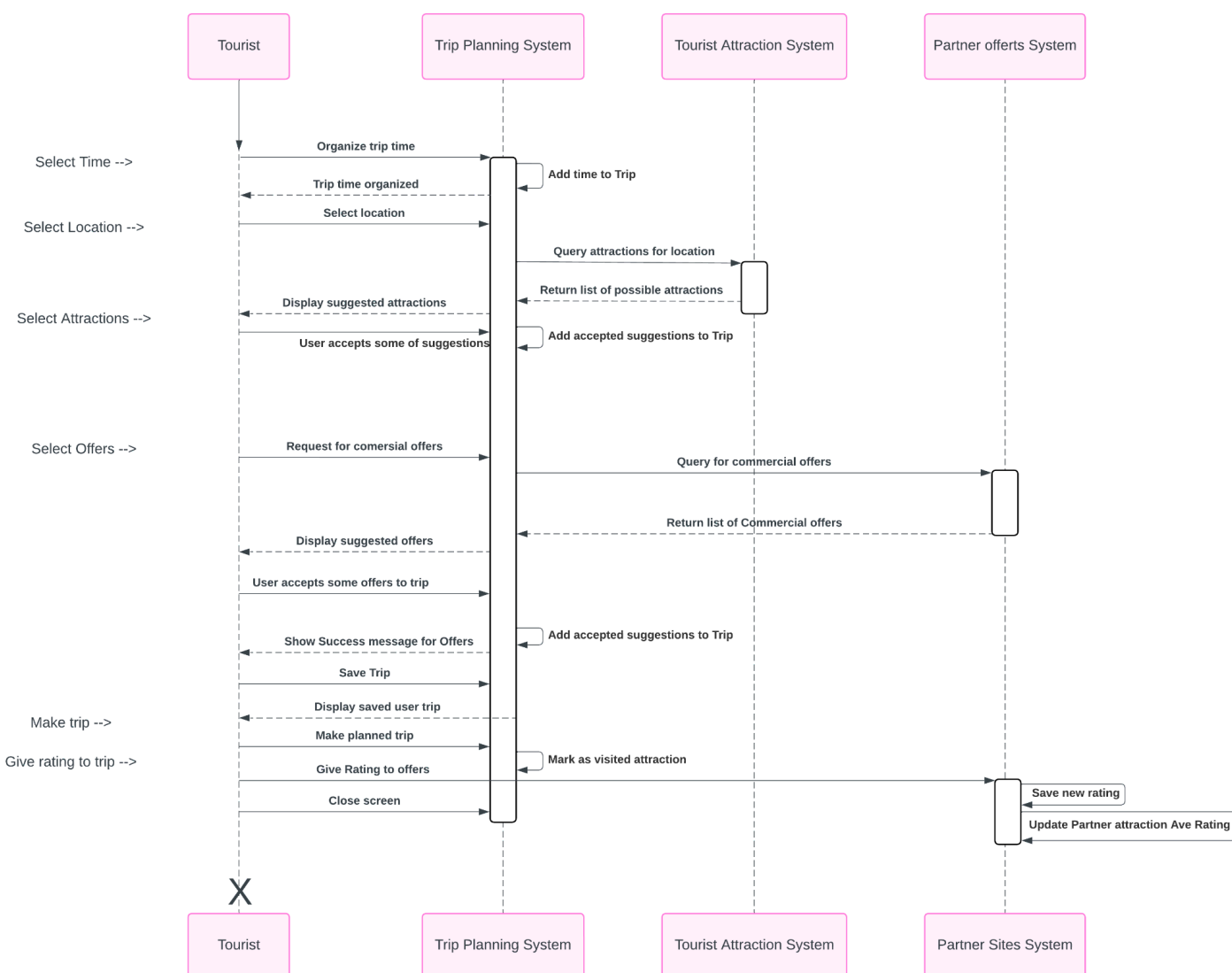
7.1. Вписване на потребителя



7.2. Планиране на пътуване

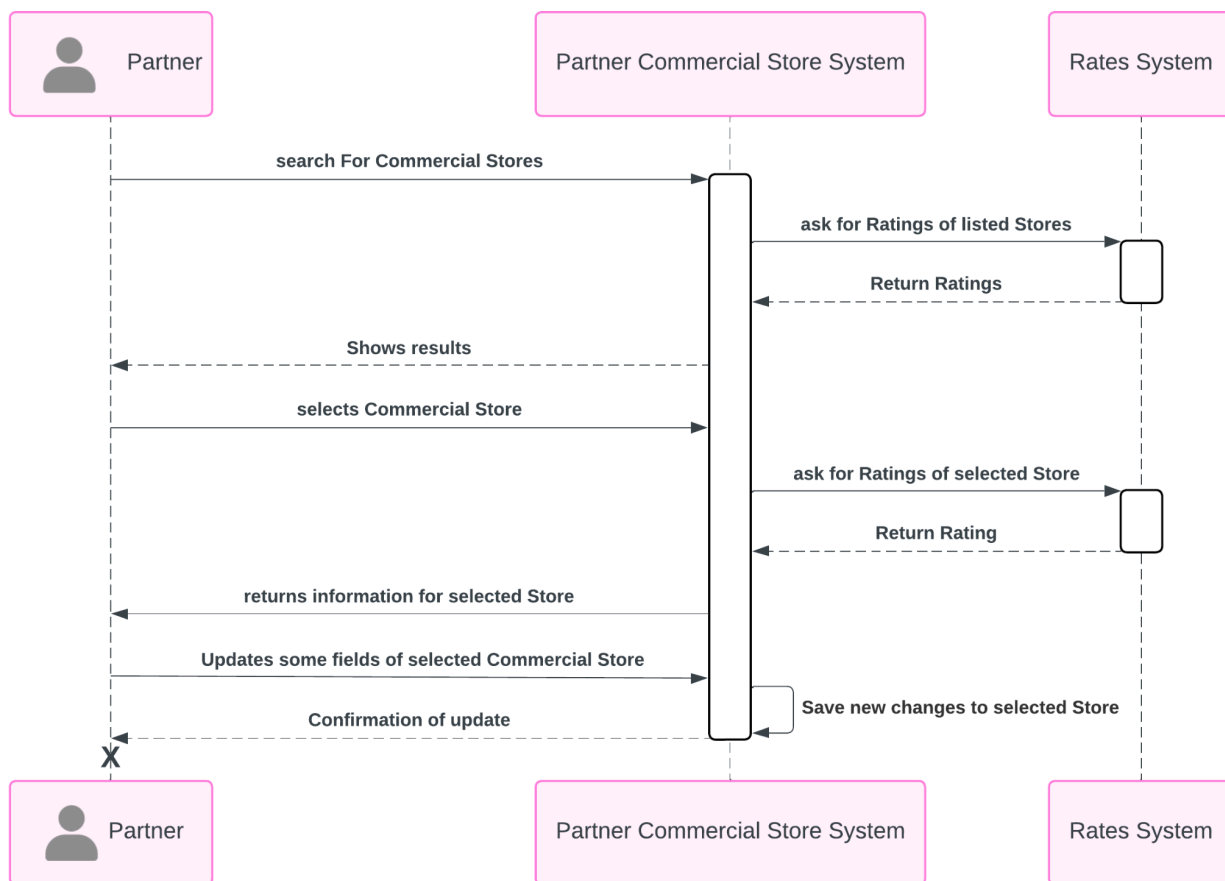
Туриста планира пътуване като премине през следните стъпки:

- Избиране на времеви интервал
- Избиране на дестинацията
- Избиране на Атракции, като получава предложения от системата за туристически обекти
- Избиране на Обекти, като получава предложения от системата за партньорски обекти
- Дава оценка на посетените атракции и обекти, след посещение.



7.3. Разглеждане и обновяване на обект

Партньорът търси свои обекти, разглежда оценките им и променя информацията за даден обект.

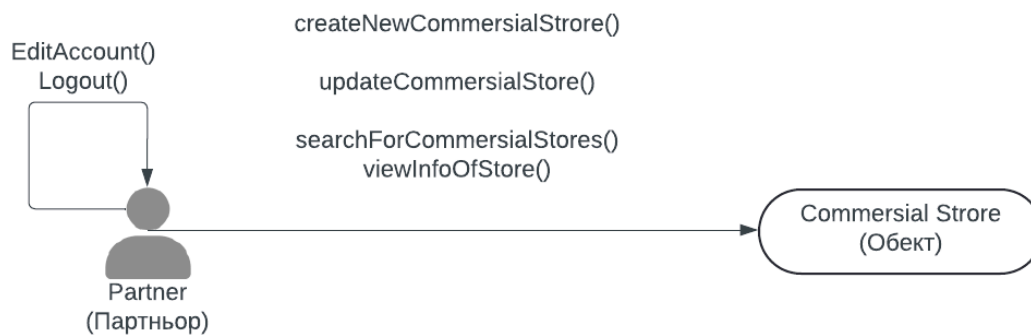


8. Communication диаграми

8.1. Посетител

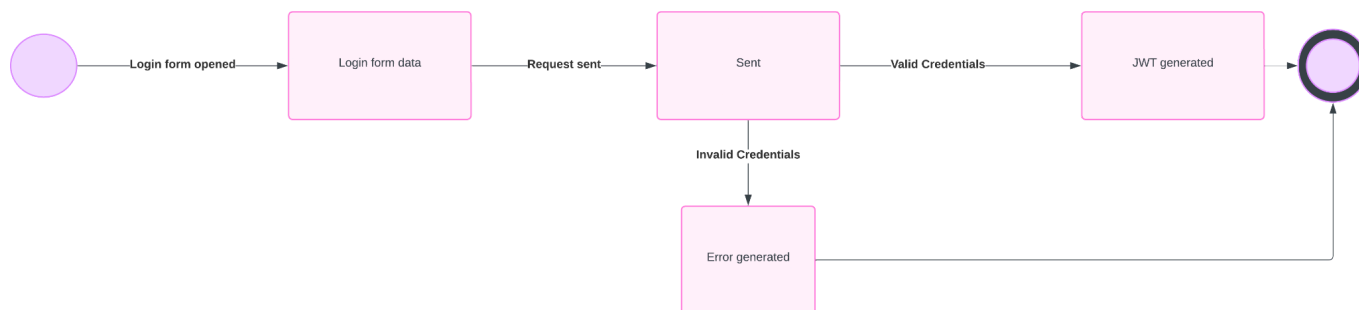


8.2. Партньор

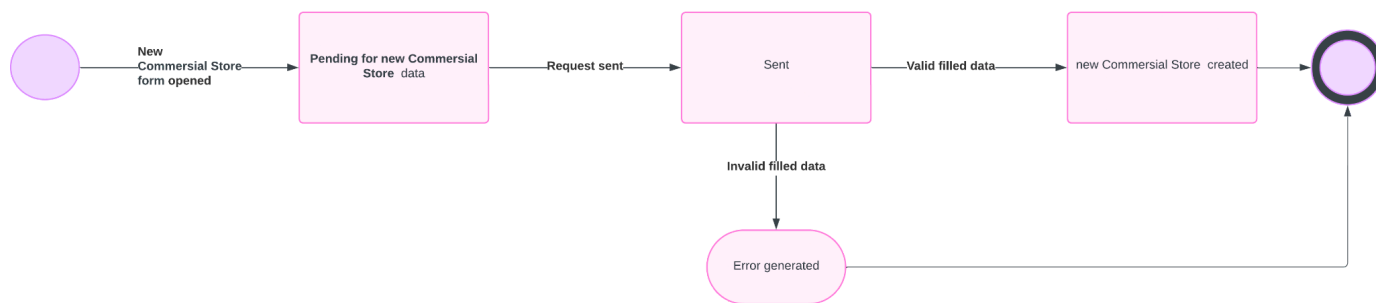


9. StateMachine

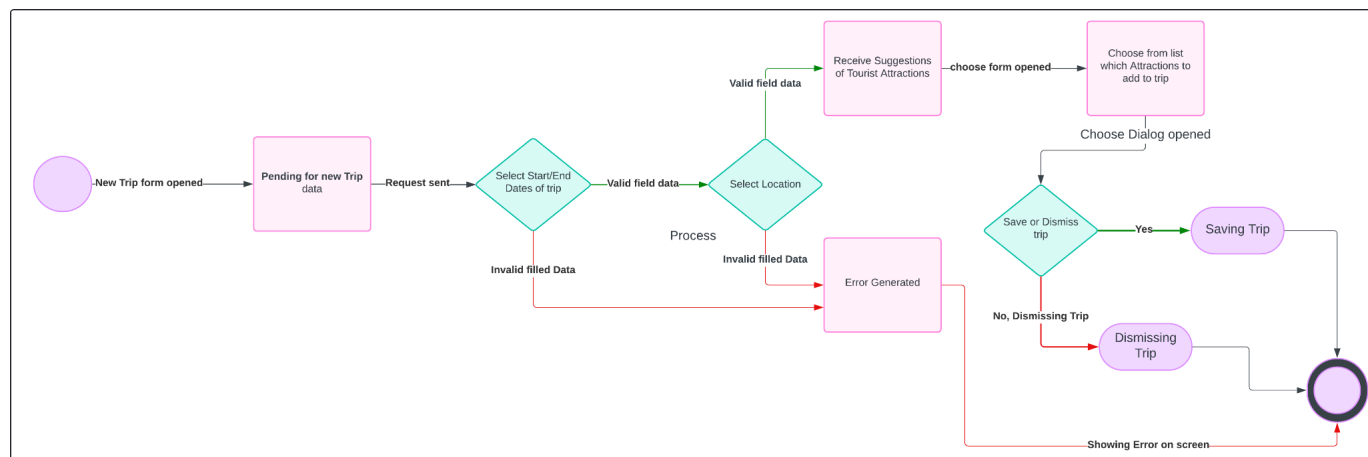
9.1. Вписване



9.2. Създаване на Партньорски Обект

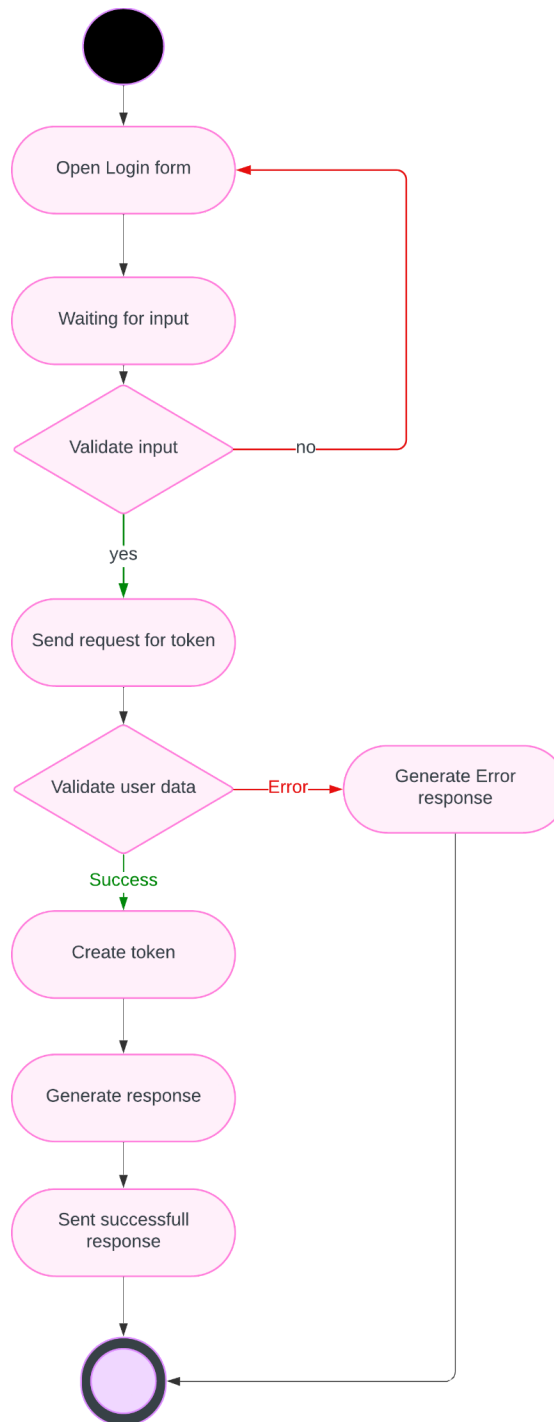


9.3. Създаване на ново пътуване като Турист

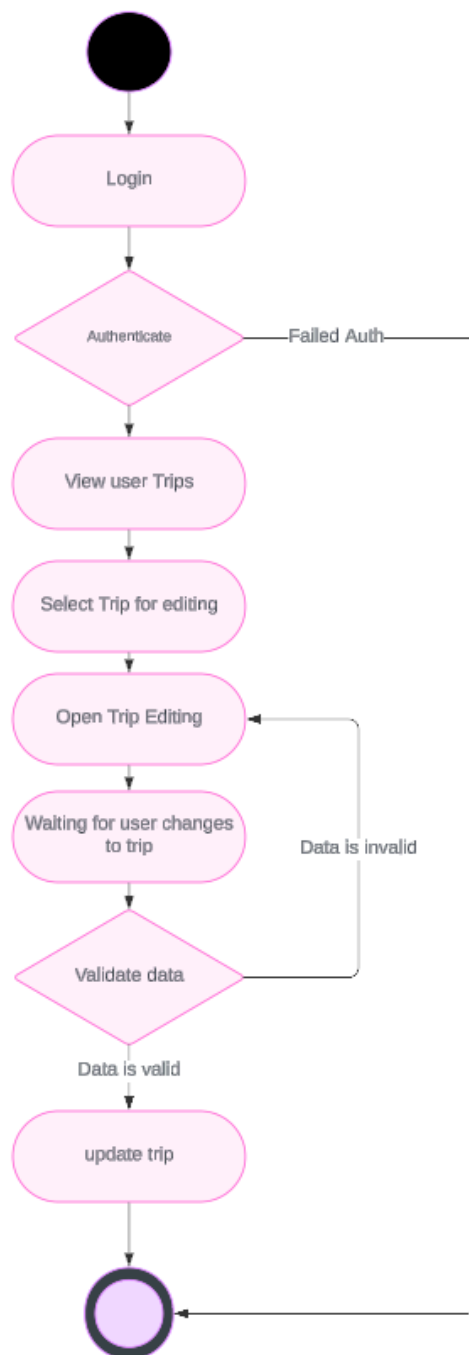


10. Activity диаграми

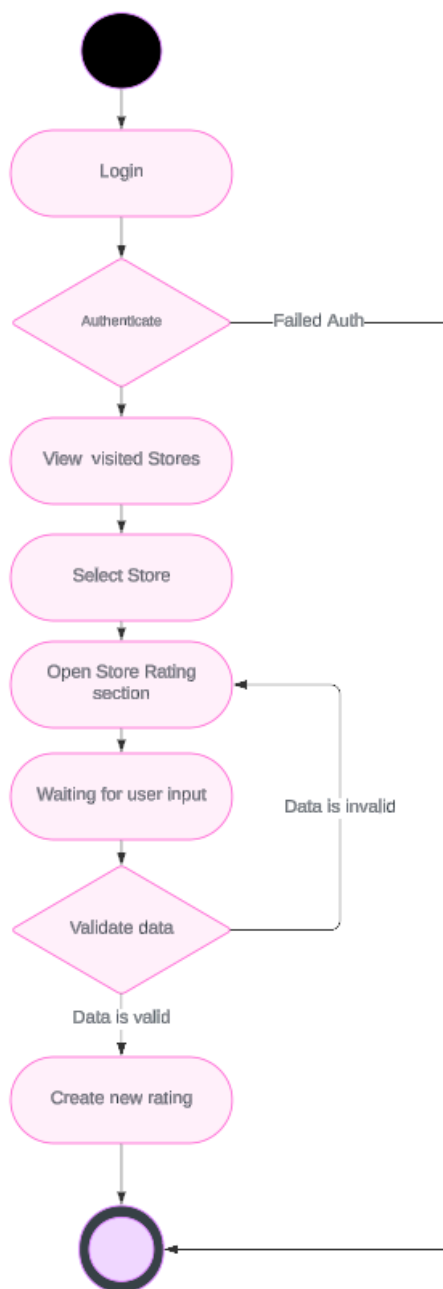
10.1. Вписване



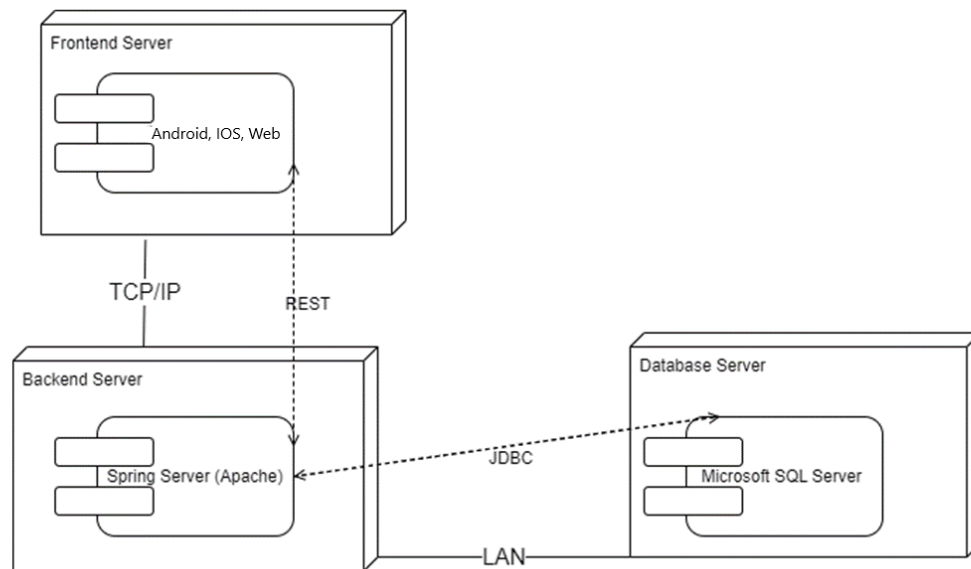
10.2. Промяна на пътуването на турист



10.3. Добавяне на оценка на партньорски обект



11. Deployment + Package диаграма



12. Използвана литература:

- [1] What is Kotlin multi-platform, Kotlin multi-platform mobile and Compose multi-platform? - [Sunish Rai](https://medium.com/@sunishrai/what-is-kotlin-multi-platform-kotlin-multi-platform-mobile-and-compose-multi-platform-b2b23d96bac8)
<https://medium.com/@sunishrai/what-is-kotlin-multi-platform-kotlin-multi-platform-mobile-and-compose-multi-platform-b2b23d96bac8>
- [2] Софтуерно проектиране и UML ООП: Упражнение 1
<https://www.malunski.com/софтуерно-проектиране-и-uml-ооп/>
- [3] Софтуерно проектиране и UML ООП: Упражнение 2
<https://www.malunski.com/софтуерно-проектиране-и-uml-ооп/>
- [4] Софтуерно проектиране и UML ООП: Упражнение 3
<https://www.malunski.com/софтуерно-проектиране-и-uml-ооп/>
- [5] Софтуерно проектиране и UML ООП: Упражнение 4
<https://www.malunski.com/софтуерно-проектиране-и-uml-ооп/>
- [6] Софтуерно проектиране и UML ООП: Примерна курсова работа
- [7] The Ultimate Guide to Sequence Diagrams - [Creately](https://medium.com/thousand-words-by-creately/the-ultimate-guide-to-sequence-diagrams-a78e0e516886)
<https://medium.com/thousand-words-by-creately/the-ultimate-guide-to-sequence-diagrams-a78e0e516886>