

Software Project

Assignment 2: Plan, design, deploy. DEEP Backlog and Product Design

Project: TourGuide Manager

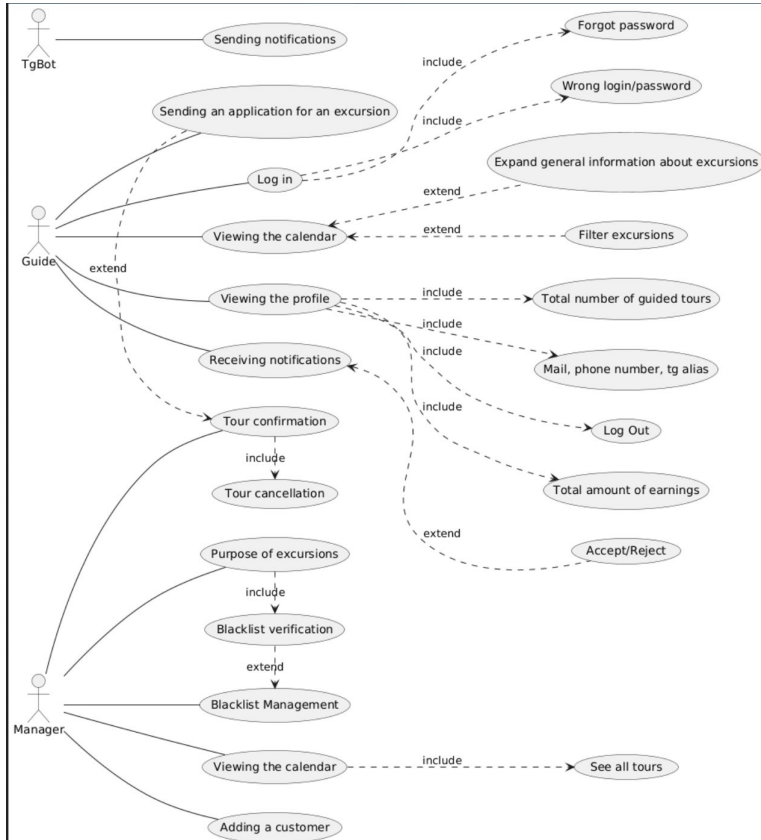
Customer: Karina Shavaliyeva (@K_Arbyzova)

Team 59 (Members and Their Contribution):

- **Alexey Chegaev** (a.chegaev@innopolis.university) - Team lead, designed and programmed interface.
- **Aleksandr Medvedev** (a.medvedev@innopolis.university) - flutter developer, learned Flutter and Dart, programmed interface.
- **Nikita Shankin** (n.shankin@innopolis.university) - flutter developer, learned Flutter and Dart, programmed interface.
- **Georgii Beliaev** (g.beliaev@innopolis.university) - web-developer, learned react, started writing web-site for admins.
- **Alexander Simonov** (a.simonov@innopolis.university) - web-developer, learned react, started writing web-site for admins.

Date: 15.06.2025

Use Case Diagram - To understand the interaction of guides and managers with excursions, and with each other



PlantUML
Code:

```

@startuml
left to right direction
actor Guide as Guide
actor Manager as Manager
actor TgBot as TgBot

usecase "Log in" as Login
usecase "Log Out" as Logout
usecase "Forgot password" as PasRes
usecase "Wrong login/password" as LogError
usecase "Expand general information about excursions" as ViewCalendarFull
usecase "Viewing the calendar" as ViewCalendarA
usecase "Viewing the calendar" as ViewCalendarA
usecase "See all tours" as SeeAllInfo
usecase "Filter excursions" as FilterTours
usecase "Viewing the profile" as ViewProfile
usecase "Total amount of earnings" as TotalSum
usecase "Total number of guided tours" as TotalExc
usecase "Mail, phone number, tg alias" as TotalInf
usecase "Sending an application for an excursion" as SendTour
usecase "Receiving notifications" as GetNotifications
usecase "Accept/Reject" as AccRej
usecase "Sending notifications" as SendNotifications
usecase "Adding a customer" as AddCustomer
usecase "Tour confirmation" as ConfirmTour
usecase "Tour cancellation" as CancelTour
usecase "Purpose of excursions" as AssignTour
usecase "Tour cancellation" as CancelTour
usecase "Blacklist Management" as ManageBlacklist
usecase "Blacklist verification" as CheckBlacklist

Guide --> Login
Guide --> ViewCalendarA
Guide --> ViewProfile
Guide --> SendTour
Guide --> GetNotifications

Login --> PasRes : include
Login --> LogError : include
ViewProfile --> TotalInf : include
ViewProfile --> TotalSum : include
ViewProfile --> TotalExc : include
ViewProfile --> Logout : include
ViewCalendarA --> FilterTours : extend
ViewCalendarA --> ViewCalendarFull : extend
GetNotifications --> AccRej : extend

Manager --> ConfirmTour
Manager --> AssignTour
Manager --> ManageBlacklist
Manager --> AddCustomer
Manager --> ViewCalendarA

ViewCalendarA --> SeeAllInfo : include
SendTour --> ConfirmTour : extend
AssignTour --> CheckBlacklist : include
ConfirmTour --> CancelTour : include
CheckBlacklist --> ManageBlacklist : extend

TgBot --> SendNotifications
@enduml
  
```

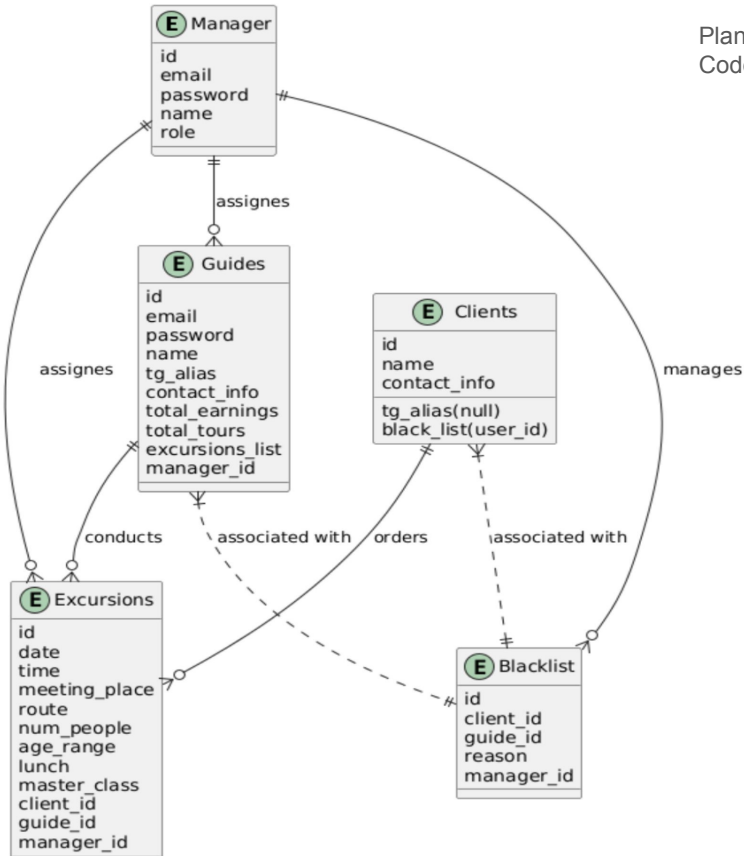
User Stories and DEEP Product Backlog

Using the Use Case Diagram, we can create user stories (backlog) displaying all 15 PBIs:

Features list	User story ID	User story
Authentication	CL-1	As a user I want to log in into my account to save history
	CL-2	As a user I want to have personal profile to see all personal info there
	CL-3	As a user I want to have ability to restore my password to log in if I forget it
	CL-4	As a user I want to specify personal info while creating account to see it in my profile
	CL-5	As a user I want to log out of the account to log in another one
Calendar	CL-6	As a user I want to see all excursions for specified date to better plan my time
	CL-7	As a user I want to see all details about excursion to be prepared to it
	CL-8	As a user I want to see 14 future days in calendar to check my future excursions
Admin interface	CL-9	As an admin I want to have calendar to assign excursions to guides
	CL-10	As an admin I want to have blacklist of guides to avoid assigning excursions to guides I don't want to
	CL-11	As an admin I want to have personal account to have personal blacklist of users
	CL-12	As an admin I want to delete assigned excursion in case it was cancelled
Telegram bot	CL-13	As a user I want to get notifications about new excursions in telegram to claim/reject them
	CL-14	As a user I want to open the app in telegram to have quick access
	CL-15	As a user I want to see response to my action of confirming or rejecting excursion to understand if I got it or not

Using the illustrative example of the Use Case Diagram, it is also necessary to create an ER diagram to understand how the database works during all processes:

Data Model (ERD)



PlantUML
Code:

```

@startuml
skinparam entityBorderColor #000000
skinparam entityFontName Arial
  
```

```

entity Manager {
    id
    email
    password
    name
    role
}
  
```

```

entity Guides {
    id
    email
    password
    name
    tg_alias
    contact_info
    total_earnings
    total_tours
    excursions_list
    manager_id
}
  
```

```

entity Excursions {
    id
    date
    time
    meeting_place
    route
    num_people
    age_range
    lunch
    master_class
    client_id
    guide_id
    manager_id
}
  
```

```

    client_id
    guide_id
    manager_id
}
  
```

```

entity Clients {
    id
    name
    tg_alias(null)
    contact_info
    black_list(user_id)
}
  
```

```

entity Blacklist {
    id
    client_id
    guide_id
    reason
    manager_id
}
  
```

```

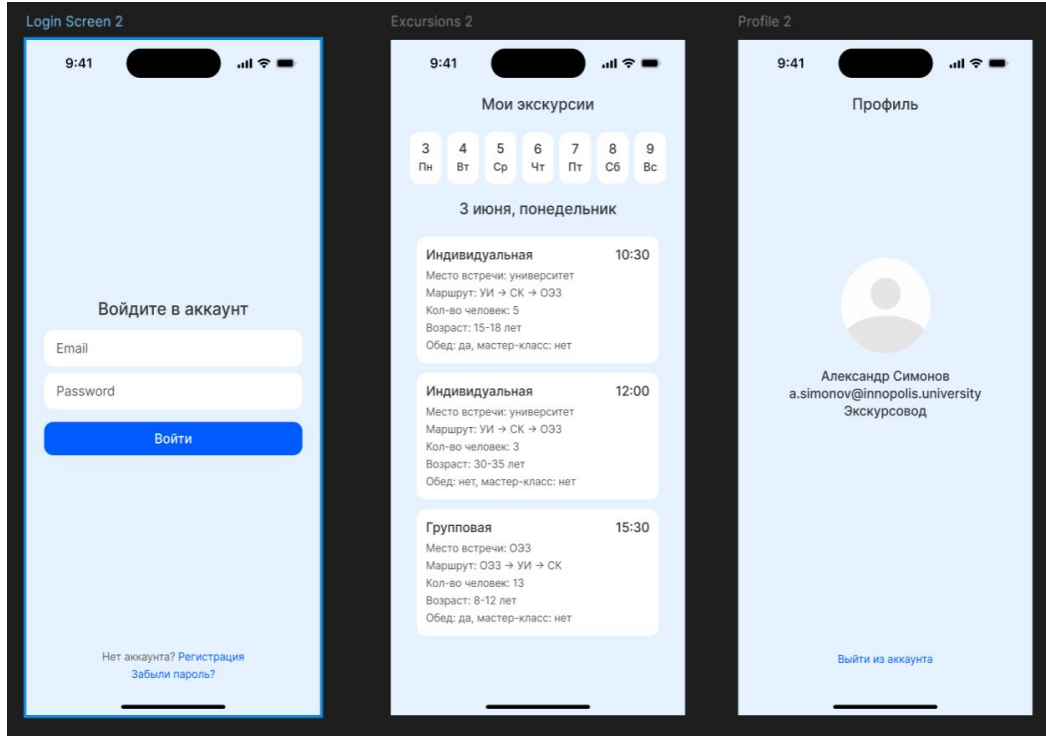
Manager }--o{ Guides : "assignes"
Manager }--o{ Excursions : "assignes"
Manager }--o{ Blacklist : "manages"
  
```

```

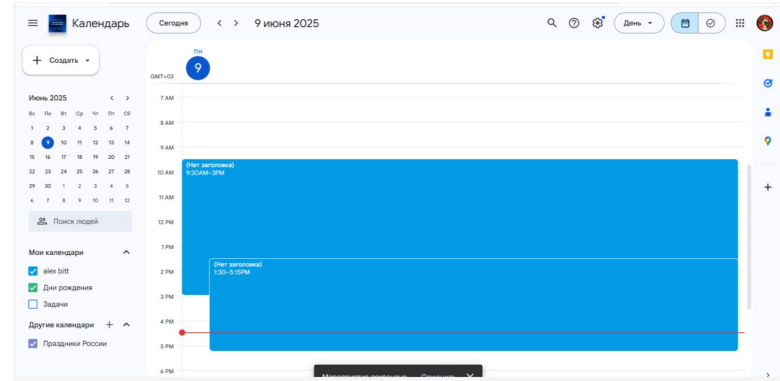
Guides }--o{ Excursions : "conducts"
Clients }--o{ Excursions : "orders"
Clients }..|| Blacklist : "associated with"
Guides }..|| Blacklist : "associated with"
@enduml
  
```

UI design

App (for Guides)



Web-site (for Admins)



Links to Figma:

- Main:
<https://www.figma.com/design/g2XK5KFh1hFa3twEZwEhHO/Untitled?t=i79zD5WrRloDypH8-1>
- Design Options:
<https://www.figma.com/design/OqBlgS0PRcCHLKHjv35nHv/Tour-App?node-id=0-1&t=iQXQYMtG4iLXBcjU-1>

MVP 0:

- Figma: <https://www.figma.com/design/g2XK5KFh1hFa3twEZwEhHO/Untitled?t=j79zD5WrRloDypH8-1>
- GitHub: https://github.com/ProjectSWD12/SWD_Project
- GitLab: <https://gitlab.pg.innopolis.university/a.medvedev/swp>
- PostMan (API):
<https://www.postman.com/aleksandr-5724752/tour-notification-api/collection/00bgsuo/tour-notification-api>

Activity Tracking Sheet:

https://docs.google.com/spreadsheets/d/1ZRA1j2vE8hXVuOt5l8hp_lct9rOE9e6-tlsH4S-XGns/edit?gid=1793499882#gid=1793499882

Key feedback points from the customer:

- It is necessary to use one common database (CRM) for storing and synchronizing excursions.
- We agreed to set a low priority for the task of sending notifications via the tg bot.
- The calendar must be synchronized with the guide's calendar.
- It is necessary to take into account the wishes of the clients regarding the guides.
- The manager does not schedule excursions, he opens them on request, and the guides themselves must have time to take the tour (the difference between excursions should be 30 minutes).

AI: we used AI tool DeepSeek so that he could help us create an API, understand the concept of DEEP, and also provide code for PlantUML, which we later fixed and supplemented.