**User Stories:**

**Sprint 1 (22.3 – 19.4):**

1. As a resident, I can sign up and create a user profile in order to connect to the application.
   1. Task analysis.
   2. Create corresponding models.
   3. Create backend sign up system.
   4. Create back-end authentication system.
   5. Create corresponding view (UI).
   6. Create front-end and back-end validation system.
   7. Create redirection from this action.
   8. Test plan.
   9. Test creation.
   10. Test execution.
2. As a resident, I can sign in to B7Fun application with my details (email and password) in order to enjoy the Resident application services.
   1. Task analysis.
   2. Create corresponding models.
   3. Create back-end sign in system.
   4. Create back-end authentication system.
   5. Create back-end session system.
   6. Create corresponding view (UI).
   7. Create front-end and back-end validation system.
   8. Create redirection from this action.
   9. Test plan.
   10. Test creation.
   11. Test execution.
3. As a resident, I can update my user details in order to keep my profile updated and relevant.
   1. Task analysis.
   2. Create corresponding models.
   3. Create corresponding back-end logic.
   4. Create corresponding view (UI).
   5. Create front-end and back-end validation system.
   6. Test plan.
   7. Test creation.
   8. Test execution.
4. As a resident, I can retrieve my password (by sending a retrieve password to my email) in order to retrieve a forgotten password and continue to enjoy B7Fun services.
   1. Task analysis.
   2. Create corresponding models.
   3. Create corresponding back-end mail system.
   4. Create corresponding back-end password retrieve system.
   5. Create corresponding view (UI).
   6. Create redirection from this action.
   7. Test plan.
   8. Test creation.
   9. Test execution.
5. As an admin, I can sign in to B7Fun application with my details (a pre-created super user) in order to use B7Fun services..
   1. Task analysis.
   2. Create corresponding models.
   3. Create back-end sign in system.
   4. Create back-end authentication system.
   5. Create back-end session system.
   6. Create corresponding view (UI).
   7. Create front-end and back-end validation system.
   8. Create redirection from this action.
   9. Test plan.
   10. Test creation.
   11. Test execution.
6. As an admin, I can manage all databases through B7Fun application in order to maintain all databases easily and make sure all is up to date.
   1. Task analysis.
   2. Create corresponding database.
   3. Create back-end manage database logic.
   4. Create corresponding view (UI).
   5. Create front-end and back-end validation system.
   6. Test plan.
   7. Test creation.
   8. Test execution.
7. As an admin, I can change my password in order to secure the B7Fun admin section.
   1. Task analysis.
   2. Create corresponding models.
   3. Create corresponding back-end logic.
   4. Create corresponding UI.
   5. Create redirection from this action.
   6. Test plan.
   7. Test creation.
   8. Test execution.
8. As an admin, I can log out from B7Fun application in order to keep my B7Fun data and services safe.
   1. Task analysis.
   2. Create corresponding back-end logic.
   3. Create corresponding session control.
   4. Create corresponding UI.
   5. Test plan.
   6. Test creation.
   7. Test execution.
9. As admin, I can manage (add or delete) posts to my posts page in order to notify users about situations.
   1. Task analysis.
   2. Create corresponding models.
   3. Create back-end manage posts logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
10. As a resident, I can log out to B7Fun in order to keep my data and services safe.
    1. Task analysis.
    2. Create corresponding back-end logic.
    3. Create corresponding session control.
    4. Create corresponding UI.
    5. Test plan.
    6. Test creation.
    7. Test execution.
11. As a resident, I can view my profile in order to check my user profile data.
    1. Task analysis.
    2. Create corresponding models.
    3. Create corresponding back-end logic.
    4. Create corresponding view (UI).
    5. Create redirection from this action.
    6. Test plan.
    7. Test creation.
    8. Test execution.

**Sprint 2 (20.4 – 17.5):**

1. As a resident, I can view list of all sport and fun activities spots in Be'er Sheva so I can see the locations relevant to me for my enjoyment.
   1. Task analysis.
   2. Create corresponding models.
   3. Create corresponding back-end logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
2. As a resident, I can view a map of all sport and fun activities spots in Be'er Sheva so I can see the locations relevant to me for my enjoyment.
   1. Task analysis.
   2. Create corresponding models.
   3. Create corresponding back-end logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
3. As a resident, I can select a location from the list and get more information on that location in order to expand my knowledge and see if location is relevant to me.
   1. Task analysis.
   2. Create corresponding models.
   3. Create corresponding back-end logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
4. As a resident, I can search locations in order to get results relevant to me.
   1. Task analysis.
   2. Create corresponding models.
   3. Create back-end search system logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
5. As a resident, I can filter locations based on pre-created filter in order to get results relevant to me.
   1. Task analysis.
   2. Create corresponding models.
   3. Create back-end filter system logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
6. As a resident, I can view the current weather in Be'er Sheva in order to plan my activities based on the weather.
   1. Task analysis.
   2. Create corresponding models.
   3. Create back-end weather system logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
7. As an admin, I can send a distribution email to all user in order to notify users about a situation.
   1. Task analysis.
   2. Create corresponding models.
   3. Create back-end send mail system logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
8. As an admin, I can send a specific email to a user in order to notify that specific user about a situation.
   1. Task analysis.
   2. Create corresponding models.
   3. Create back-end send mail system logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
9. As an admin, I can view my last actions in order to track my administration action in the admin section.
   1. Task analysis.
   2. Create corresponding models.
   3. Create corresponding back-end logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
10. As an admin, I can view users review of B7Fun application in order to improve B7Fun..
    1. Task analysis.
    2. Create corresponding models.
    3. Create corresponding back-end logic.
    4. Create corresponding view (UI).
    5. Test plan.
    6. Test creation.
    7. Test execution.
11. As an admin, I can view the B7Fun resident application in order to check and validate site functionality.
    1. Task analysis.
    2. Create corresponding models.
    3. Create corresponding back-end logic.
    4. Create corresponding view (UI).
    5. Test plan.
    6. Test creation.
    7. Test execution.

**Sprint 3 (18.5 – 7.6):**

1. As an admin, I can send a user an email with a warning about him using abusive content in order to maintain order.
   1. Task analysis.
   2. Create corresponding models.
   3. Create back-end send mail system logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
2. As a resident, I can send a review to admin about the B7Fun application in order to help improve the application.
   1. Task analysis.
   2. Create corresponding models.
   3. Create back-end send review system logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
3. As a resident, I can send admin alerts about services and locations in order to help maintain sites data and B7Fun application.
   1. Task analysis.
   2. Create corresponding models.
   3. Create back-end send alerts system logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
4. As resident, I can join a chat room of a specific location in order to speak with people that this location is relevant to them as well.
   1. Task analysis.
   2. Create corresponding models.
   3. Create back-end and front-end chat system logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
5. As a resident, I can send report abusive messages inside chat rooms in order to maintain a healthy environment in B7Fun application.
   1. Task analysis.
   2. Create corresponding models.
   3. Create back-end report abusive messages logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
6. As a resident, I can view the other user’s profile in order to learn about people in my community.
   1. Task analysis.
   2. Create corresponding models.
   3. Create corresponding back-end logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
7. As an admin, I can view messages marked as abusive in order to deal with that content correctly.
   1. Task analysis.
   2. Create corresponding models.
   3. Create back-end mark as abusive system logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
8. As an admin, I can block users from the system in order to maintain order.
   1. Task analysis.
   2. Create corresponding models.
   3. Create back-end clock user system logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
9. As an admin, I can view number of users who signed in to B7Fun in a certain period in order to get information about application usage.
   1. Task analysis.
   2. Create corresponding models.
   3. Create corresponding back-end logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.
10. As an admin, I can view number of users who signed up to B7Fun in a certain period in order to get information about application popularity.
    1. Task analysis.
    2. Create corresponding models.
    3. Create corresponding back-end logic.
    4. Create corresponding view (UI).
    5. Test plan.
    6. Test creation.
    7. Test execution.

**Extra user stories (optional):**

1. As a Resident, I can view locations in my favorite list in order to view them later faster without searching for them again.
   1. Task analysis.
   2. Create back-end get favorite system logic.
   3. Create corresponding view (UI).
   4. Test plan.
   5. Test creation.
   6. Test execution.
2. As a Resident, I can add locations to my favorite list in order to maintain a list of relevant locations for me.
   1. Task analysis.
   2. Create back-end add favorite system logic.
   3. Create corresponding view (UI).
   4. Test plan.
   5. Test creation.
   6. Test execution.
3. As an admin, I can view which sites or most read by user in order to follow popular sites.
   1. Task analysis.
   2. Create corresponding models.
   3. Create corresponding back-end logic.
   4. Create corresponding view (UI).
   5. Test plan.
   6. Test creation.
   7. Test execution.