|  |  |
| --- | --- |
| Fully-dressed description | |
| Use Case UC-1 | Attain health data |
|
| Related requirements :  Initiating actor:  Actor’s Goal:  Partitioning Actors:  Preconditions:  Postconditions: | ST\_9, ST\_13, ST\_15  Users  Collecting the user’s data, analysis and evaluations  Database, health information API  user is logged in with proper authorization from the system  health information API is updated with user’s modification |
| Flow of events for main successful scenario | |
|  | 1. user successfully log in to the system 2. system requires information from health API 3. system display the required information 4. user reviews the information and modify the information 5. system notify the API and updates the information |
| Use Case UC-3 | Get Recommendation from the healthy diet |
| Related requirements :  Initiating actor:  Actor’s Goal:  Partitioning Actors:  Preconditions:  Postconditions: | ST\_11  Users  To hear suggestions from the system  diet recommendation API, Database  user is logged in with proper authorization from the system  Recommendations are kept in the system |
| Flow of events for main successful scenario | |
|  | 1. user successfully log in to the system   2 . user asks for recommendation  3. system requires diet API to provide recommendation  4. system display the recommendation  5. user determine whether to follow the suggestions  6 . system collect the reflection from the user  7. system determine whether to keep the recommendation based on the user’s decision |
| Use Case UC-6 | Share the information to social website and invited friends |
| Related requirements :  Initiating actor:  Actor’s Goal:  Partitioning Actors:  Preconditions:  Postconditions: | ST\_16  Users  To share the information to others  Sharing API, other platform, Database  user is logged in and the internet access is available  Information from others is kept in the system |
| Flow of events for main successful scenario | |
|  | 1. user successfully log in to the system 2. system requires information from health API 3. system display the required information 4. user reviews the information and modify the information 5. system notify the API and updates the information |

|  |  |
| --- | --- |
| Use Case UC-8 | Customize own diet plan |
|
| Related requirements :  Initiating actor:  Actor’s Goal:  Partitioning Actors:  Preconditions:  Postconditions: | ST\_10  Users  Modify the recommended diet plan  Database, diet recommend API  user has at least ask the system to recommend healthy diet  diet API is updated with user’s modification |
| Flow of events for main successful scenario | |
|  | 1. user successfully log in to the system 2. user require the recommendation diet from system 3. system display the required diet 4. user reviews the diet and modify the diet 5. system notify the API and updates the diet |
| Use Case UC-9 | View others health data and goal with permission |
| Related requirements :  Initiating actor:  Actor’s Goal:  Partitioning Actors:  Preconditions:  Postconditions: | ST\_6, ST\_7, ST\_8  Users  Compare with friends to share experience  Database, social communication API  both the user and user’s friend are using the system  the database would keep user’s friend’s health information |
| Flow of events for main successful scenario | |
|  | 1. user successfully log in to the system 2. user send a request to his friend to get health information 3. user get permission 4. system attain required information through network connection 5. system display the attained information 6. database keep the health information of the user’s friend |
| Use Case UC-11 | Get a recommended wearable equipment |
| Related requirements :  Initiating actor:  Actor’s Goal:  Partitioning Actors:  Preconditions:  Postconditions: | ST\_12  Users  Get recommendation of wearable equipment  Database, wearable equipment recommend API  User logged in and database has health information about the user  wearable equipment API is updated, database keep the information |
|  | 1. user successfully log in to the system 2. user requires recommendation of wearable equipment 3. system look into the database 4. based on the information stored in the database, the recommendation API make suggestions 5. system display the recommendation 6. user determine whether to follow the suggestion |

|  |  |
| --- | --- |
| Use Case UC-13 | Get feedback |
|
| Related requirements :  Initiating actor:  Actor’s Goal:  Partitioning Actors:  Preconditions:  Postconditions: | ST\_17  Web maintainer  To achieve feedback from users  Database, feedback API  user has made feedback  feedback is kept in the database |
| Flow of events for main successful scenario | |
|  | 1. user made feedback 2. the database keeps the feedback information 3. web maintainer requires the feedback information 4. system display the feedback information |
| Use Case UC-14 | Update database and modify UI |
| Related requirements :  Initiating actor:  Actor’s Goal:  Partitioning Actors:  Preconditions:  Postconditions: | ST\_19  Web maintainer  To update database and UI  Database,  database is not empty  database is updated in the latest version |
| Flow of events for main successful scenario | |
|  | 1. database collect information from the user 2. web maintainer log in to the database 3. web maintainer require to update the database and modify UI 4. the database and UI keeps the latest updates and modification |