**a.Domain Model**

**i. Concept Definitions**

To analyze the domain model, we first derive the domain model concepts and corresponding responsibilities from the formerly defined system use cases. Table lists all the domain model concepts and corresponding responsibilities

|  |  |  |
| --- | --- | --- |
| **Responsibility Description** | **Type** | **Concept Name** |
| R1:Store the data about healthy information | K | Database |
| R2:Read user’s custom diet plan | D | DietReader |
| R3:Read data that user types in | D | TextReader |
| R4:Container for user’s custom diet plan | K | DietPlan |
| R5:Controller receives the operation “Custom Diet Plan” and controls DietReader to get DietPlan and send to DB Connection to write into the Database. | D | Controller |
| R6: HTML document that shows the user’s healthy data | K | UserPage |
| R7:HTML document that shows the user’s healthy data such as number of steps , sleeping time and activity hours to analyst. | K | AnalystPage |
| R8: Prepare a database query that best matches the actor’s input and retrieve the records from the database | D | DB connection |
| R9:Connect to Social Network and retrieve information | D | SN connection |
| R10: Render the retrieved records into an HTML document for sending to actor’s Web browser for display. | D | Page Maker |
| R11: Controller receives the operation “Share on Facebook” and controls SN connection to connect to social network and retrieve information | D | Controller |
| R12: Controller receives the operation “Get advice” and controls DB connection to get Advice from Database | D | Controller |
| R13:Container for the advice that analyst gives | K | Advice |
| R14:Checker receives the username and requests Controller to use DB connection for using datatbase(InfoStorage) to find data in database to compare. | D | Checker |
| R15:Read advice from analysts | K | AdviceReader |
| R16: Controller receives the operation “Input advice” and controls AdviceReader to get Advice and send to DB Connection to write into the Database. | D | Controller |
| R17: Checker gets information from TextReader and check if the information is valid | D | Checker |

**ii. Association definitions**

Some of the concepts defined above as domain concepts have to work in certain patterns to finish some target requirements. Table below gives the corresponding association definitions based on the defined domain concepts.

|  |  |  |
| --- | --- | --- |
| **Concept Pair** | **Association Description** | **Association Name** |
| Controller<->Checker | Controller calls checker to check if information is valid and Checker returns results to Controller or checker requests to use database and controller returns results | Generate requests  Convey data |
| TextReader<->Controller | TextReader sends signals to Controller or Controller sends signals to TextReader to receive data | Generate Requests |
| Controller<->Page Maker | Controller renders its results and generate requests to Page Maker to display | Generate requests  Convey data |
| Checker<->TextReader | Checker checks information from TextReader | Check |
| Checker<->Database | Check checks information from Database | Check |
| DB Connection<->Database | DB Connection gets access into Database and saves data in Database | Save data |
| Controller<->DB Connection | Controller generates requests to use DB connection | Render requests |
| Controller<->Advice Reader | Controller calls Advice Reader to read Advice from analyst | Generate requests  Convey data |
| Checker<->DietReader | Checker checks information from DietReader | Check |
| Controller<->SN Connection | Controller generates requests to use SN connection | Render requests  Convey data |

**iii. Attribute Definition**

Among the defined concepts, some concepts share the same attribute, and only different from each other as they have different operands. These concepts are listed in below.

|  |  |  |
| --- | --- | --- |
| **Concept** | **Attributes** | **Attribute Description** |
| TextReader | Read input data | Allow actor inputs data and put the external data into system |
| DietReader |
| AdviceReader |
| Database | Data storage | Store all the data that the system needs |
| Controller | ReceiveInfo  NoticeChecker | Control the check to validate data |
| NoticePageMaker | Control the page maker to generate page to display data |
| NoticeSNConnection | Control the SN connection to connect social network and get information from it |
| NoticeAdviceReader | Control the advice reader to read advice and store in database |
| NoticeDietReader | Control the diet plan reader to read diet plan and store in database |
| UserPage | Display Result | Interface for user and analyst to interact with system |
| AnalystPage |
| Checker | Validate data | Check whether the data is legal |
| PageMaker | Generate New Page | Generate Userpage or Analystpage to show the result or get input |
| SN Connection | Connect to Social Network | Connect to social network to share information or retrieve information |

iv. Traceability Matrix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Domain Model | UC-1 | UC-3 | UC-6 | UC-8 | UC-9 | UC-11 |
| Database | ✓ | ✓ | ✓ | ✓ | ✓ |  |
| DietReader |  |  |  | ✓ |  |  |
| TextReader | ✓ |  |  |  |  |  |
| AdviceReader |  | ✓ |  |  |  | ✓ |
| Controller | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| UserPage | ✓ |  |  |  | ✓ | ✓ |
| AnalystPage |  | ✓ |  |  |  | ✓ |
| PageMaker | ✓ | ✓ |  |  | ✓ | ✓ |
| Checker | ✓ | ✓ |  | ✓ |  |  |
| DBConnection | ✓ | ✓ | ✓ | ✓ | ✓ |  |
| SNConnection |  |  | ✓ |  | ✓ |  |

**b.System Operation Contracts**

**Attain health data**

Preconditions: user is logged in with proper authorization from the system

Postconditions: health information API is updated with user’s modification

**Get Recommendation from the healthy diet**

Preconditions: user is logged in with proper authorization from the system

Postconditions: Recommendations are kept in the system

Share the information to social website and invited friends

Preconditions: user is logged in and the internet access is available

Postconditions: Information from others is kept in the system

**Customize own diet plan**

Preconditions: user has at least ask the system to recommend healthy diet

Postconditions: diet API is updated with user’s modification

**View others health data and goal with permission**

Preconditions: both the user and user’s friend are using the system

Postconditions: the database would keep user’s friend’s health information

**Get a recommended wearable equipment**

Preconditions: User logged in and database has health information about the user

Postconditions: wearable equipment API is updated, database keep the information