SE Unit2 Project – YADA (Yet Another Diet Assistant)

Team 3

- Bhavana Gannu
- Padmalata Nistala
- Arpita Gupta
- Ranjith Raj
- AVS Murthy

Agenda

1. YADA– Requirements
2. Design – UML Class Diagram
3. Classes - Responsibilities
4. Design - Salient Points
5. Implementation

1. YADA – Functional Requirements

- Food Items
  - Add, update, Delete basic food items with basic attributes – identifier, keywords, calories (UI)
  - Add, update, delete composite food items with basic attributes (UI)
  - Create composite food by selecting one or more basic or composite food items with one or more qty.
  - Ability to search for basic or composite food item based on key words provided
  - Load basic and composite food items from database files on program start
  - Save basic and composite food items at program exit
  - Download food items information from standard web sites

- User Profile
  - Create user profile with basic information
  - Maintain age, weight, activity level information on a daily basis for user
  - Update age, weight, activity level information on a day by user as needed
  - Selection of target calorie intake method by User

1. YADA – Functional Requirements

- User Food Log
  - Maintain log of food consumed on a daily basis for user
  - Load user food log from database for a specific user
  - Add, delete entries from food logs for a specific date by selecting food items through search
  - By default carry over previous day's log record for current day.
  - Undo option for user commands for log activities

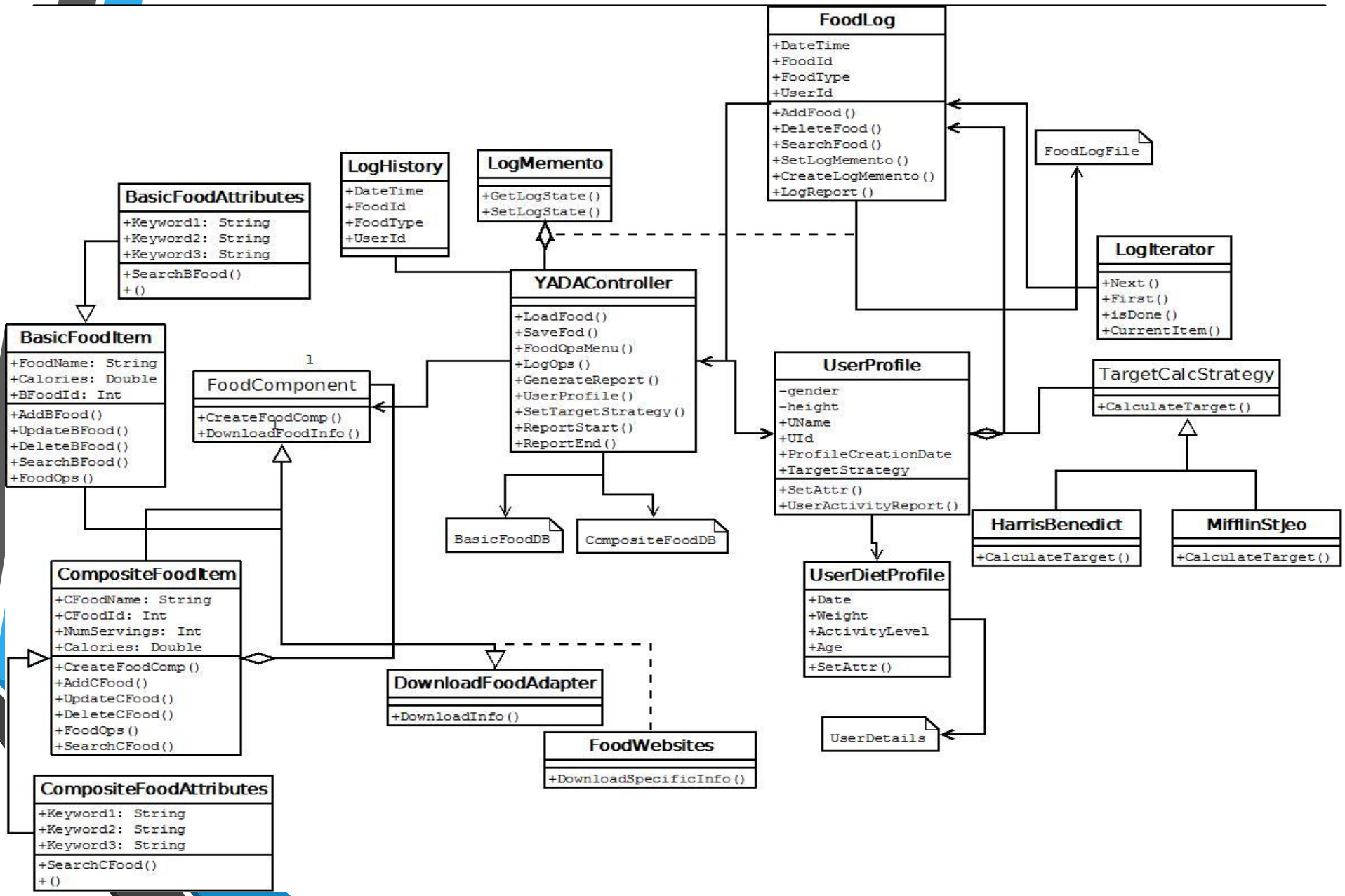
- Reporting
  - Generate day wise food log report for a user for a time period (from date, to date)
  - Format of day wise food log report – User, date, total calorie consumed, target calorie intake, difference between target and consumed.

- Maintain databases
  - Basic food items
  - Composite food items
  - User profile
  - User daily food log

2. YADA – Non Functional Requirements

- Maintainability
  - To be able to add additional attributes to basic and composite food items (e.g. nutrition information)
- Interoperability
  - Read and write food and log data to text files (csv)
  - Ability to extend for downloading of food information from web sites
- Adaptability
  - To be able to choose target calorie intake method for calarie computation
- Performance
  - Managing size of log files
- Security
  - User registration feature – to be added (later)
  - Accountability of transaction create, update, delete (later)

3. Design – UML Diagram



3. Classes - Responsibilities

Class	Responsibility
YADAController	Main controller class, which gets invoked first during execution -Provides the user interface to select options -Initiate Load files from databases, Save files to database, Initiate generate report
UserProfile	Provides basic profile of user
UserDietProfile	Provides Diet profile of user for a day – weight, activity level…
Food Component	**Composite pattern** – can have basic or composite food items as food components
Basic Food Item and Basic Food Attrs	Basic food items and corresponding attributes
Composite Food Item and Composite Food Attrs	Composite food items and corresponding attributes
Download food adapter and Food Websites	**Adapter pattern -** To provide interface for creating food items components from food webs sited through DownloadFoodAdapter.
Food Log	Maintains log of food consumed by user – date wise
Log Iterator	**Iterator pattern** – To traverse user log collection
Target Calc Strategy,	**Strategy pattern** – To facilitate selection of target calorie intake method by User
HarrisBenedict, MifflinStJoe	Specific algorithm implementations
LogMemento, LogHistory	**Memento pattern -** To provide undo option for user commands(log)

4. Design - Salient Points

Pattern	Responsibility
Composite Pattern	Food component creation-can have either basic or composite food items.
Strategy Pattern	To allow for selection of algorithm for target calorie computation and to have flexibility to add more algorithms to design at a later point.
Adapter Pattern	To have flexibility to add interfaces for downloading food information from websites and populating to food items.
Iterator Pattern	To traverse the log collection for a user for a specific date.
Memento Pattern	To provide facility to undo log operations by users

Target Calorie Algorithms	Description
HarrisBenedict	A concrete calculator that calculates target calorie intake using Miffin-St Jeo Algorithm.
MifflinStJoe	A concrete target calorie calculator. Uses Harris-Benedict Algorithm.

4. Design - Salient Points

- **Databases -** Maintained as csv text files
  - UserFoodDB
  - BasicFoodDB
  - CompositeFoodDB
  - UserLogDB - this file is maintained user wise and year wise - to manage the size of the log file.

- **Strengths**
  - Leveraging of appropriate design patterns (Strategy and Adapter) to provide flexibility and maintainability.
  - Tried for high cohesion and loose coupling in class design
  - Traceability of identified functional and non functional requirement in design

- **Weakness**
  - Security requirements identified but not implemented for prototype. Security is weak and needs to be built into the design.
  - Primitive user interface is provided from command prompt. Can be improved.

5. Implementation

Programming Language: C++

- Database: Text Files

```
UsersFile users;

UserProfile* check(string username,string password)
{
    vector<UserProfile> userlist = users.getUserList();
    vector<UserProfile>::iterator it;

    for(it=userlist.begin();it!=userlist.end();++it)
    {

        if(it->getUserName().compare(username)==0)
        {
            return  new UserProfile(it->getUserName(),it->getGender(),it->getAge(),it->getWeight(),it->getHeight(),it->getActivityLevel());
        }

    }

    return NULL;
}
```

Thank You