CS5704 Software Engineering

Semester Project Report

Fall 2021

**HealthTech**

Ayush M. Shah

Divyansh S. Mishra

Neel C. Gada

Department of Computer Science

Virginia Tech

Blacksburg, VA 24061

*App's URL:* /HealthTech-Team10/

*Date:* October 4th, 2021

*Team Number:* 10

*Instructor:* Prof. Osman Balci

USER ACCOUNTS

Using the deployed software, each team member is required to create an account and generate *meaningful content* in the database for evaluation and testing. **You will be penalized for not having sufficient content**.

In the case of two-factor authentication, your software is required to provide a Bypass Two-Factor Authentication option for grading.

List the usernames and passwords below to use for grading the app.

|  |  |
| --- | --- |
| *Username* | *Password* |
| Team Member 1's username | Team Member 1's password |
| Team Member 2's username | Team Member 2's password |
| Team Member 3's username | Team Member 3's password |

This page must be numbered as ii. No page number must appear on the title page.

EXECUTIVE SUMMARY

TBD

\*\*\*\*

**ATTENTION**: Delete all instructions in red before submitting the final version of the report.

===> 5 points out of 100 will be deducted if any instruction is left in the final version. <===

\*\*\*\*

Provide a structured summary of the content of the document here by emphasizing your important contributions.

**List of Implemented Cloud Software Features Learned**

1. TBD
2. TBD
3. TBD
4. TBD
5. TBD
6. TBD
7. TBD
8. TBD
9. TBD
10. TBD
11. TBD
12. TBD
13. TBD
14. TBD
15. TBD

**List of Implemented New Cloud Software Features**

1. TBD
2. TBD
3. TBD
4. TBD
5. TBD

This page must be numbered as iii.

**TABLE OF CONTENTS**

[USER ACCOUNTS ii](#_Toc79596689)

[EXECUTIVE SUMMARY iii](#_Toc79596690)

[1. SOFTWARE LIFE CYCLE 1](#_Toc79596691)

[2. OUR OBJECTIVE 2](#_Toc79596692)

[3. PROBLEM SPECIFICATION 2](#_Toc79596693)

[3.1 What is the problem? 2](#_Toc79596694)

[3.2 Why is the problem important to solve? 2](#_Toc79596695)

[3.3 Expected Functionality Description 2](#_Toc79596696)

[3.4 Learned and new cloud software features to be implemented 2](#_Toc79596697)

[4. REQUIREMENTS SPECIFICATION 4](#_Toc79596698)

[4.1 Functional Requirements 4](#_Toc79596699)

[4.2 Non-Functional Requirements 4](#_Toc79596700)

[5. ARCHITECTURE SPECIFICATION 5](#_Toc79596701)

[5.1 OV-1: High-Level Operational Concept Graphic 5](#_Toc79596702)

[5.2 OV-2: Operational Resource Flow Description 5](#_Toc79596703)

[5.3 OV-5b: Operational Activity Model 6](#_Toc79596704)

[5.4 SV-1: Systems Interface Description 6](#_Toc79596705)

[5.5 SV-2: Systems Resource Flow Description 6](#_Toc79596706)

[5.6 SV-4: Systems Functionality Description 6](#_Toc79596707)

[5.7 SvcV-1: Services Context Description 6](#_Toc79596708)

[5.8 SvcV-2: Services Resource Flow Description 6](#_Toc79596709)

[5.9 SvcV-4: Services Functionality Description 6](#_Toc79596710)

[6. DESIGN SPECIFICATION 7](#_Toc79596711)

[7. DELIVERED SOFTWARE FUNCTIONALITY 8](#_Toc79596712)

[8. CONCLUSIONS 9](#_Toc79596713)

[9. SUBMISSION INSTRUCTIONS 9](#_Toc79596714)

[10. PERCENTAGES OF CONTRIBUTION 10](#_Toc79596715)

[11. CALCULATION OF GRADES BASED ON PERCENTAGES OF CONTRIBUTION 11](#_Toc79596716)

[12. TEAM PROJECT REQUIREMENTS 11](#_Toc79596717)

[13. GRADING SHEET 12](#_Toc79596718)

[REFERENCES 13](#_Toc79596719)

[APPENDIX A: MEETING MINUTES 14](#_Toc79596720)

Each entry in the TOC above must be a hyperlink clickable to jump to that section in the report.

Right click the TOC above and select **Update Field**. Select **Update entire table** in the dialog box.

You must update the TOC to reflect changes before submitting the final version.

This page must be numbered as iv.

# SOFTWARE LIFE CYCLE

A good software engineer develops software by following the software life cycle shown below.



A programmer (hacker or ad-hoc developer) develops software by looking at the problem and directly coding in an IDE. This approach is known as the *Build-and-Fix Approach*, which must never be used!

This page must be numbered as 1 and subsequent pages must be numbered accordingly

# OUR OBJECTIVE

The objective of our team project is to **demonstrate how capable the team members are** in engineering a Jakarta/Java EE cloud software application to solve a complex problem. The cloud software application is created for the purpose of showing how learned and new complex functionalities and features the team members are capable of developing.

# PROBLEM SPECIFICATION

## What is the problem?

Obesity has been on a rise in US over the last couple of decades. As per the CDC, from 2000 to 2018, obesity prevalence increased from 30.5% to 42.4%, leading to increase in related health conditions like heart disease, type-2 diabetes, and strokes. People have easy access to unhealthy food and often struggle to moderate their intake or balance it out with adequate physical activity.

While workout-management applications exist, they hardly ever provide sophisticated tools to track macronutrient level intake. If and when they do, it is either not comprehensive enough or blocked behind a hefty paywall. Similar is the case for diet-management applications. Applications that provide both workout management and diet monitoring features are few and far between and are usually designed around expensive subscriptions.

With HealthTech, our goal is to help individuals lead a healthier life by offering a web-app to track their macronutrient intake and calories burned while exercising powered by sophisticated visualization tools, all for the charge of $0 (yes you read it right, its FREE!)

## Why is the problem important to solve?

There is an increased awareness amongst people now to lead a healthier and active life. Diet control and physical activity monitoring is an important step in this direction. However, most people avoid seeing a nutrition expert due to high cost associated with it. This is usually followed by doing research about health/exercise/weight-loss online and an ad-hoc plan to reach one’s goal. To follow a healthier diet, it is important to measure calorie intake, as it is hard to improve something you don’t measure. It can be overwhelming to do so much research and manually track food intake and calories burned. There is also the chance of falling for one of the many health-fads being peddled online. Lack of results also lead most people to give up early.

In this scenario, it is important to have a software that automates diet and exercise tracking using sophisticated visualization tools with a $0 subscription fee. Such a tool will help people lead healthier lives and reach and surpass their health goals.

## Expected Functionality Description

Diagram

Description automatically generated

## Learned and new cloud software features to be implemented

Our proposed app shall implement the following *learned* and *new* cloud software features.

Table 1. List of Learned Cloud Software Features to be Implemented

|  |  |
| --- | --- |
| *No.* | *Description of the Learned Cloud Software Feature to be Implemented* |
| 1 | User can create account and sign into app with email and password |
| 2 | User can reclaim account if they forget password via recovery code emailed to email account used at sign up |
| 3 | Display list of common food items with their calories and macronutrient breakdown, with ability to filter by column |
| 4 | Create food item and provide the macronutrients as input |
| 5 | Users can input food items consumed and see calories consumed along with macronutrient breakdown |
| 6 | User can set goal of how many calories they want to consume and see daily progress towards goals |
| 7 | User can design workouts and track calories burned. |
| 8 | Users can input workouts done and see it for future reference |
| 9 | User can query database from app for all food items falling in a desired range |

Table 2. List of New Cloud Software Features to be Implemented

|  |  |
| --- | --- |
| *No.* | *Description of the New Cloud Software Feature to be Implemented* |
| 1 | Generate weekly report of calories consumed with visualization and analysis |
| 2 | Recommend food items and workouts to user as per goals set in app |
| 3 | User can bookmark favorite food items |
| 4 | Share goals achieved on social media |
| 5 | Get complete nutritional breakdown of any recipe |

# REQUIREMENTS SPECIFICATION

This section specifies the Functional and Non-Functional Requirements under which our Jakarta/Java EE cloud software application will be developed.

## Functional Requirements

1. TBD
2. TBD
3. TBD
4. ...

Minimum 18 correct Functional Requirements are required. (6 per team member)

## Non-Functional Requirements

Each team shall satisfy the following non-functional requirements:

1. The UI template footer shall contain the following statement:  
     
   [CS5704 Software Engineering](https://manta.cs.vt.edu/cs5704/) course semester project application developed by studentName1, studentName2, and studentName3.  
     
   Course title and each student name above shall be hyperlinked to show the corresponding homepage *in a new window / tab*. Use VT website if you do not have one for yourself.
2. The cloud software application name in IntelliJ IDEA and when deployed on the server shall be a meaningful name without course number or team number. The app name shall reflect what the app does such as MeetingScheduler. Do **not** name it as “group”, “Team4App”, or “CS5704Team4Project”.

Each team’s non-functional requirements:

1. TBD
2. TBD
3. TBD
4. ...

Minimum 9 correct Non-Functional Requirements are required. (3 per team member)

# ARCHITECTURE SPECIFICATION

TBD

You are required to describe the network-centric / cloud-based **architecture** of your software-based solution system using the following DoDAF models:

|  |  |
| --- | --- |
| Team Member 1 shall create the following 3 DoDAF Models | |
|  | 1. **OV-1**: High-Level Operational Concept Graphic 2. **OV-2**: Operational Resource Flow Description 3. **OV-5b**: Operational Activity Model |
| Team Member 2 shall create the following 3 DoDAF Models | |
|  | 1. **SV-1**: Systems Interface Description 2. **SV-2**: Systems Resource Flow Description 3. **SV-4**: Systems Functionality Description |
| Team Member 3 shall create the following 3 DoDAF Models | |
|  | 1. **SvcV-1**: Services Context Description 2. **SvcV-2**: Services Resource Flow Description 3. **SvcV-4**: Services Functionality Description |

**Notes:**

* Study [DoDAF Volume 2](https://manta.cs.vt.edu/cs5704/StudentsOnly/Handouts/DoDAFv2Vol2.pdf) to learn about each type of DoDAF model (diagram).
* See [Examples of DoDAF models for a Campus Situational Awareness and Emergency Response Management System](https://manta.cs.vt.edu/sinergy/).
* **Microsoft Visio** (recommended) or a tool of your choice can be used to create the DoDAF diagrams.
* Each student is required to create 3 DoDAF models (diagrams) as indicated above. A team of 2 members shall create the models for Team Members 1 and 2 or 1 and 3 or 2 and 3.
* **Extra credit** will be given for additional DoDAF diagrams. Extra credit will be determined with respect to the complexity of each extra diagram.

## OV-1: High-Level Operational Concept Graphic

TBD

## OV-2: Operational Resource Flow Description

TBD

## OV-5b: Operational Activity Model

TBD

## SV-1: Systems Interface Description

TBD

## SV-2: Systems Resource Flow Description

TBD

## SV-4: Systems Functionality Description

TBD

## SvcV-1: Services Context Description

TBD

## SvcV-2: Services Resource Flow Description

TBD

## SvcV-4: Services Functionality Description

TBD

# DESIGN SPECIFICATION

TBD

This section shall present the UI design and Database design with annotated graphics.

|  |  |
| --- | --- |
| **Software Tools for User Interface and Navigation Structure Prototyping** | |
|  | * Balsamiq, <https://balsamiq.com/> (recommended) * Marvel App, <https://marvelapp.com/> * Framer, <https://framer.com/> * Proto.io, <https://proto.io/> |

|  |  |
| --- | --- |
| **Software Tool for Relational Database Design** | |
|  | * DbDesigner.net, <https://dbdesigner.net/> |

If your app uses a file storage directory, name it as CS5704-TeamN-FileStorage, where N is your team number.

# DELIVERED SOFTWARE FUNCTIONALITY

TBD

Describe all of the functionalities (features) of your deployed Java EE cloud software application by using screenshots of user interfaces. For each UI screenshot, describe how the software functions.

Start with the URL for accessing your software.

Then, screenshot by screenshot, describe how your software is used.

All of the software functionalities (features) must be described in sufficient detail so that your project can be properly graded.

# CONCLUSIONS

TBD

# SUBMISSION INSTRUCTIONS

1. If your app uses a file storage directory, name it as   
   CS5704-TeamN-FileStorage, where N is your team number.
2. Deploy your cloud software application to your AWS EC2 virtual private server computer.
3. Using the deployed software, each team member is required to create an account and generate *meaningful content* in the database for evaluation and testing. You will be penalized for not having sufficient content. In the case of two-factor authentication, your software is required to provide a Bypass Two-Factor Authentication option for grading. List the usernames and passwords under this option in the *User Accounts* section on page ii.
4. Open your IntelliJ project and change the file paths for your app to run locally.
5. Create a ZIP file containing the following:
   1. This file: *CS5704 Team N Semester Project Report.docx* (Replace N with your team number. Do not change this filename and do not remove the spaces in the filename!)
   2. Your IntelliJ IDEA Ultimate project folder containing your entire documented application.
   3. Your Database SQL file to populate the tables in your database.
   4. Your CS5704-TeamN-FileStorage directory, original version with no data.
6. Upload the ZIP file to your account in Google Drive, right click the uploaded file and select **Share** from the pop-up menu, enter [balci@vt.edu](mailto:balci@vt.edu), add a note as “CS5704 Team N Semester Project Submission – Your Name”, and click Send.

# PERCENTAGES OF CONTRIBUTION

We hereby certify that the list of contributions and the corresponding percentages of contribution specified below truly reflect the actual contributions of the team members.

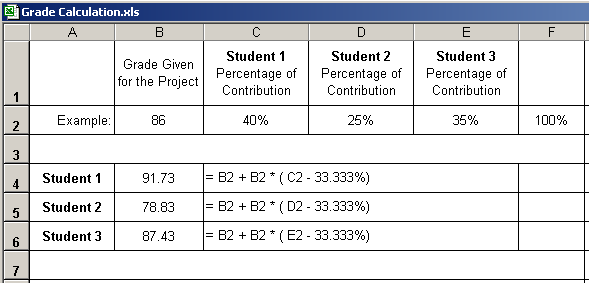
(Write your name as your signature)

|  |  |  |
| --- | --- | --- |
| *Student Name & List of Contributions* | *% Contributed* | *Signature* |
| Student Name 1’s Contributions:   1. tbd 2. tbd 3. tbd 4. ... | a% |  |
| Student Name 2’s Contributions:   1. tbd 2. tbd 3. tbd 4. ... | b% |  |
| Student Name 3’s Contributions:   1. tbd 2. tbd 3. tbd 4. ... | c% |  |
| Sum of Percentages: | 100% |  |

The Virginia Tech Honor Code is fully in effect for the above declaration. The percentage of contribution must be justified by the list of contributions specified in detail.

If you disagree with your team members, you and the other team members must separately email Dr. Balci and indicate what each member thinks the fair percentages of contribution are for all team members.

# CALCULATION OF GRADES BASED ON PERCENTAGES OF CONTRIBUTION



**Additional percentage of contribution cannot exceed 10%.**

**If a situation arises where a student is doing more than 10% extra work, Dr. Balci must be informed immediately.**

# TEAM PROJECT REQUIREMENTS

1. Each team member is expected to contribute equally.
2. You shall submit percentages of contribution together with a list of each student’s individual contributions with signatures of all team members. In case of disagreement, you shall submit it separately with your rationale. (Write your name as representing your signature.)
3. Grades shall be determined based on the percentages of contribution.
4. If you contribute more than your equal share, then your grade shall be increased based on the extra percentage of contribution, which cannot be more than 10%.
5. If you contribute less than your equal share, then your grade shall be decreased accordingly.
6. The extra percentage of contribution shall not be more than 10%. Dr. Balci shall be notified immediately if a situation arises where a student needs to contribute more than 10% extra.
7. A team member who does not cooperate with other team members for conducting the project with equal contribution shall be penalized. Doing more work than agreed upon by the team and claiming extra contribution shall not be acceptable. Cooperation is essential!

# GRADING SHEET



REFERENCES

TBD

Balci, O. (2021), “CS3754 Cloud Software Development Course Website,” <https://manta.cs.vt.edu/cs3754>

Eclipse Foundation (2021), “Jakarta EE,” <https://jakarta.ee/>

JetBrains (2021), “Intelligent Java Integrated Development Environment Advanced (IntelliJ IDEA),” <https://www.jetbrains.com/idea/>

MySQL (2021), “MySQL Open Source Relational Database Management System,” <http://www.mysql.com/>

PrimeTek (2021), “PrimeFaces UI Component Library for JSF,” <https://www.primefaces.org/>

WildFly (2021), “WildFly Jakarta / Java Application Server,” <https://www.wildfly.org/>

<< Reference other sources used in your work. Include other references used in alphabetical order styled under the style **References** >>

APPENDIX A: MEETING MINUTES

Meeting Number 1

|  |  |
| --- | --- |
| *Date & Duration:* | 10/25/2021 |
| *Location:* | TORG 1040 |
| *Members Present:* | Ayush Shah, Divyansh Mishra, Neel Gada |
| *Members Absent:* | - |
| *Discussions:* | 1. Sketched rough layout for index page  2. Login/signup dialog box and input fields and their validations  3. Nutrient tracker along with visualization dialog box |
| *Decisions Made:* | 1. To have login and signup in a dialog box 2. To explore other options other than hamburger menu, possibly to display page navigation options in a navbar 3. Slider images stretches the entire width 4. On nutrient tracker page group meals by date and display all meals consumed in a day in a single component and have an option to view visualization in a dialog box 5. To display popular recipes in a tabular format 6. Input fields for Signup (Name, Email, Password, Confirm Password) and password should be alphanumeric with length between 8-12 characters with at least one uppercase, lowercase, and special character). 7. In login have an option to reset password by sending 6-digit integer OTP to user email address |
| *Work Assignments:* | Ayush: Login and Signup flow development along with reset password  Divyansh: Project setup, database design and calorie tracker page  Neel: Index page and list page development |
| *Minutes Prepared By:* | Ayush Shah |

Meeting Number 2

|  |  |
| --- | --- |
| *Date & Duration:* |  |
| *Location:* |  |
| *Members Present:* |  |
| *Members Absent:* |  |
| *Discussions:* |  |
| *Decisions Made:* |  |
| *Work Assignments:* |  |
| *Minutes Prepared By:* |  |

Meeting Number 3

|  |  |
| --- | --- |
| *Date & Duration:* |  |
| *Location:* |  |
| *Members Present:* |  |
| *Members Absent:* |  |
| *Discussions:* |  |
| *Decisions Made:* |  |
| *Work Assignments:* |  |
| *Minutes Prepared By:* |  |

:

:

:

(List for all meetings held)

:

:

:

Meeting Number N

|  |  |
| --- | --- |
| *Date & Duration:* |  |
| *Location:* |  |
| *Members Present:* |  |
| *Members Absent:* |  |
| *Discussions:* |  |
| *Decisions Made:* |  |
| *Work Assignments:* |  |
| *Minutes Prepared By:* |  |