**Software Design Specifications**

User API

Version 0.1

© 2021

[1 Document Control Information 4](#_Toc46771737)

[1.1 Document Control Numbering 4](#_Toc46771738)

[1.2 Version Management 4](#_Toc46771739)

[1.3 Ownership Matrix 4](#_Toc46771740)

[2 Introduction & Overview 6](#_Toc46771741)

[2.1 Introduction 6](#_Toc46771742)

[2.2 Problem Statement 6](#_Toc46771743)

[2.3 Purpose of Solution 6](#_Toc46771744)

[2.4 Acronyms & Abbreviations 7](#_Toc46771745)

[3 Stakeholder Information 8](#_Toc46771746)

[3.1 Function Requested By 8](#_Toc46771747)

[3.2 Function Users 8](#_Toc46771748)

[3.3 Stakeholder Provided Information 8](#_Toc46771749)

[3.4 Licensing & Module Dependency 9](#_Toc46771750)

[4 Solution Overview 10](#_Toc46771751)

[4.1 Solution Summary 10](#_Toc46771752)

[4.2 Scope of Work 10](#_Toc46771753)

[4.3 Pages 10](#_Toc46771754)

[4.4 Solution Strengths 10](#_Toc46771755)

[4.5 Solution Weaknesses 10](#_Toc46771756)

[4.6 Solution Opportunities 10](#_Toc46771757)

[4.7 Solution Assumptions & Limitations 10](#_Toc46771758)

[5 Documents & Other Information 10](#_Toc46771759)

[5.1 SRS – Software Requirements Document 10](#_Toc46771760)

[5.2 Reference product backlog items 10](#_Toc46771761)

[6 Detailed Explanation of Solution 10](#_Toc46771762)

[6.1 Database changes 10](#_Toc46771763)

[6.2 Schemas changes 11](#_Toc46771764)

[6.2.1 Claim type 11](#_Toc46771765)

[6.3 Rest API 11](#_Toc46771766)

[6.3.1 CreateSystemRole 11](#_Toc46771767)

[6.3.2 UpdateSystemRole 12](#_Toc46771768)

[6.3.3 RemoveSystemRole 13](#_Toc46771769)

[6.3.4 CreateUser 14](#_Toc46771770)

[6.3.5 UpdateUser 17](#_Toc46771771)

[6.3.6 RemoveUser 18](#_Toc46771772)

[6.3.7 AddUserClaims 19](#_Toc46771773)

[6.3.8 UpdateUserClaims 20](#_Toc46771774)

[6.3.9 RemoveUserClaims 20](#_Toc46771775)

[6.3.10 GetUserClaims 21](#_Toc46771776)

[6.3.11 SetUserAvatar 21](#_Toc46771777)

[6.3.12 CreateGroup 22](#_Toc46771778)

[6.3.13 UpdateGroup 22](#_Toc46771779)

[6.3.14 RemoveGroup 23](#_Toc46771780)

[6.3.15 GetGroup 23](#_Toc46771781)

[6.3.16 GetGroups 23](#_Toc46771782)

[6.3.17 AddRoleInGroup 24](#_Toc46771783)

[6.3.18 RemoveRoleFromGroup 24](#_Toc46771784)

[6.3.19 ResetUserPassword 24](#_Toc46771785)

[6.4 Database seeds 25](#_Toc46771786)

[7 Use Cases for Solution 25](#_Toc46771787)

[8 Open questions 25](#_Toc46771788)

[9 Functional Requirements (FEAT’S) 25](#_Toc46771789)

[10 Impact Summary 26](#_Toc46771790)

[10.1 Change Listing 26](#_Toc46771791)

[11 Master and Child Ticket Tracking 28](#_Toc46771792)

[11.1 Master Project Ticket 28](#_Toc46771793)

[11.2 Child Project Tickets 28](#_Toc46771794)

[12 REVIEW MOM & HISTORY 29](#_Toc46771795)

[13 Approval & Sign Off 30](#_Toc46771796)

# Document Control Information

## Document Control Numbering

|  |  |
| --- | --- |
| Sugar Case Number |  |
| Azure DevOps Issue Number |  |
| Functional Specification Number |  |
| Functional Specification Module |  |
| Functional Function Category |  |
| Distributed To |  |
| Number of Pages in Document |  |
| First Version & Date |  |

## Version Management

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Author | Date | Sections Changed |
| 0.01 | Dmytro Shakhov | 9/20/2020 | Draft |
| 0.02 |  |  |  |
| 0.03 |  |  |  |
| 0.04 |  |  |  |
| 0.05 |  |  |  |
| 0.06 |  |  |  |
| 0.07 |  |  |  |
| 0.08 |  |  |  |
| 0.09 |  |  |  |
| 0.10 |  |  |  |

## Ownership Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Department | Role | Name | Notes |
| Development | Back-end Developer | Dmytro Shakhov |  |
| Development |  |  |  |
| Development |  |  |  |
| QA |  |  |  |

# Introduction & Overview

## Introduction

This document describes the API of User functionality

## Problem Statement

We have a fictitious scenario where we‘d like to build an API to manage users. We require

the ability to create, get, and list users. Once we create users, we need the ability for the user to

create an account. This should be implemented with an API and database.

## Purpose of Solution

Implement API functions to meet requirements:

CreateUser,

GetUsersList,

GetUser,

CreateAccount,

GetAccountsList,

# Stakeholder Information

## Function Requested By

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Represents | Date | Role |
| Company |  | 09 /20 /2020 | Product Owner |
|  |  |  |  |

## Function Users

|  |  |  |  |
| --- | --- | --- | --- |
| User Name | User Role | Primary Objective in Solution | Frequency of Use |
| User 1 | admin | Create users, Update users, Delete users | Low |
| User 2 | user | Update own profile | Low |
| User 3 |  |  |  |

## Stakeholder Provided Information

Business Requirements

Should not allow more than one user with the same email address

- Allows credit for up to $1000. so if monthly salary - monthly expenses is less than

$1000 we should not create an account for the user and return an error

API requirements:

1. The following Endpoints are required:

- Create user

Fields to be provided and persist

- Name

- Email Address (Must be unique)

- Monthly Salary (Must be a positive number - for simplicity, pretend there is no such thing as income tax)

- Monthly Expenses (Must be a positive number)

- List users

- Get user

- Create an account

- Given a user, create an account

- Should take into account the business requirements above

- Up to you to decide the appropriate fields to persist here

- List Accounts

2. Should be robust and include relevant response codes for different scenarios

3. No need to implement any kind of auth for this test

## Licensing & Module Dependency

N/A

# Solution Overview

## Solution Summary

## Scope of Work

## Pages

## Solution Strengths

API call can be used in any rest clients as mobile application, web site application.

## Solution Weaknesses

n/a

## Solution Opportunities

We able manage users by appropriate API calls.

## Solution Assumptions & Limitations

No UI for create/update/delete users only through appropriate API calls

# Documents & Other Information

## SRS – Software Requirements Document

N/A

## Reference product backlog items

n/a

# Detailed Explanation of Solution

## Database changes

**Table “Users”**

This table stored Users records.

|  |  |  |
| --- | --- | --- |
| Field name | Field type | Size |
| Id | uniqueidentifier |  |
| Name | nvarchar | 4000 |
| Email | nvarchar | 255 |
| MonthlySalary | money |  |
| MonthlyExpenses | money |  |

**Table “Accounts”**

This table stored users Accounts records.

|  |  |  |
| --- | --- | --- |
| Field name | Field type | Size |
| Id | uniqueidentifier |  |
| UserId | uniqueidentifier |  |
| Credit | money |  |

Field UserId referenced to Users table.

## Schemas changes

### UserModel

Add new data type UserModel with properties

**email** – of string type for store e-mail address,

**name** – of string type for store user name,

**monthly**\_**salary** – of decimal(money in T-SQL) for store monthly salary value

**monthly**\_**expenses** – of decimal (money in T-SQL) for store monthly expenses value

### AccountModel

Add new data type AccountModel with properties

**user**\_**id** – of Guid type for store id user related to particular account record

**credit** – of decimal type for store value of credit

## Rest API

### CreateUser (Mongo)

**Application: TestProject.WebAPI**

POST **/api/MongoUsers/CreateUser**

**Parameters:**

No parameters

**Request body:**

Schema

|  |
| --- |
| {    "\_id": {},    "userId": {},    "name": "string333",    "createDate": "2021-09-29T13:20:58.793Z"  } |

**Responses:**

OK 200 (Created)

**Response body:**

|  |
| --- |
| {  “id”:”string”  } |

**Url sample:**

https://localhost:5001/api/MongoUsers/CreateUser

### 6.3.2. Get all users

**Application: TestProject.WebAPI**

POST **/api/MongoUsers/GetAll**

**Parameters:**

No parameters

**Request body:**

Empty

**Responses:**

OK 200

**Response body:**

|  |
| --- |
| {   {          "\_id": {              "timestamp": 0,              "machine": 0,              "pid": 0,              "increment": 0,              "creationTime": "1970-01-01T00:00:00Z"          },          "userId": {              "timestamp": 0,              "machine": 0,              "pid": 0,              "increment": 0,              "creationTime": "1970-01-01T00:00:00Z"          },          "name": "string1",          "createDate": "2021-09-26T07:34:17.41Z"      },  } |

**Url sample:**

https://localhost:5001/api/MongoUsers/GetAll

### CreateUser

**Application: TestProject.WebAPI**

POST **/api/Users /CreateUser**

**Parameters:**

No parameters

**Request body:**

Schema

|  |
| --- |
| {  “name”:”string”,  “email”:”string”  “monthly\_salary”:”number”  “monthly\_expenses”:”number”  } |

**Responses:**

Code 201 (Created)

**Response body:**

|  |
| --- |
| {  “id”:”string”  } |

### GetUsersList

**Application: TestProject.WebAPI**

GET **/api/Users/GetUsersList**

**Parameters:**

No parameters

**Responses:**

Code 200 (Success)

**Response body:**

Schema

|  |
| --- |
| {  [  {  “id”: ”string”,  “name”:”string”,  “email”:”string”,  “monthly\_salary”:”number”,  “monthly\_expenses”:”number”  },  {  “id”: ”string”,  “name”:”string”,  “email”:”string”,  “monthly\_salary”:”number”,  “monthly\_expenses”:”number”  }  ]  } |

### GetUser

**Application: TestProject.WebAPI**

GET **/api/Users/GetUser**

**Parameters:**

|  |  |  |
| --- | --- | --- |
| Name | Type |  |
| Id | String | query |

**Responses:**

Code 200 (Success)

Code 404 (Target id not found)

**Response body:**

Schema

|  |
| --- |
| {  “id”: ”string”,  “name”:”string”,  “email”:”string”,  “monthly\_salary”:”number”,  “monthly\_expenses”:”number”  } |

### CreateAccount

**Application: TestProject.WebAPI**

POST **/api/Users /CreateAccount**

**Parameters:**

|  |  |  |
| --- | --- | --- |
| Name | Type |  |
| UserId | String | query |

**Request body:**

n/a

**Responses:**

Code 201 (Created)

**Response body:**

|  |
| --- |
| {  “id”:”string”  } |

### GetAccountList

**Application: TestProject.WebAPI**

GET **/api/Users/GetAccountList**

**Parameters:**

No parameters

**Responses:**

Code 200 (Success)

**Response body:**

Schema

|  |
| --- |
| {  [  {  “id”: ”string”,  “user\_id”:”string”,  “credit”:”number”  },  {  “id”: ”string”,  “user\_id”:”string”,  “credit”:”number”  }  ]  } |

## Database seeds

### Add test users

Create test users with parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Name | Email | MonthlySalary | MonthlyExpenses |
| 1 | Foo | Foo@example.com | 100 | 100 |
| 2 | Bar | Bar@example.com | 200 | 190 |

### Add test accounts

Create test users with parameters:

|  |  |  |
| --- | --- | --- |
|  | User\_ID | Credit |
| 1 | Foo user id | 110 |
| 2 | Bar user id | 220 |

# Use Cases for Solution

TBD

# Open questions

# Functional Requirements (FEAT’S)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REQ ID | Priority | Description | Dependency / Section Map | Comments |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Impact Summary

## Change Listing

|  |  |  |  |
| --- | --- | --- | --- |
| http://www.graphicsfuel.com/wp-content/uploads/2012/10/database-512.png | **Database Changes: Yes** | **https://s3.amazonaws.com/kinlane-productions/bw-icons/bw-gears.png** | **Services Changes : No** |
| http://www.prev.johnrice.ca/_img/user-experience150X150.png | **UX / UI Changes : no** | **https://roggasblog.files.wordpress.com/2012/10/intranet-icon.png** | **Mobile Changes : no** |
| http://interfacing.com/uploads/images/Icons/rules_icon.png | **Business Rule Changes : yes** | **http://www.fromridgetoeave.com/wp-content/uploads/2015/06/icon-versatile-300x215.png** | **Communication Changes : No** |
| http://icons.iconarchive.com/icons/martz90/circle/512/windows-8-icon.png | **Windows Code Changes : Yes** | **https://cdn1.iconfinder.com/data/icons/black_and_gold/256/installer.png** | **Installer Changes : No** |
| http://www.computerservicenaples.com/media/Linux.png | **Linux Code Changes : No** |  | **Cloud / Reporting Changes : No** |

# Master and Child Ticket Tracking

## Master Project Ticket

## Child Project Tickets

|  |  |  |  |
| --- | --- | --- | --- |
| Dev Child Ticket Number | Ticket Description | Assigned To | Status |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# REVIEW MOM & HISTORY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Review Comment | Reviewer | Status |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Item # | Action Item | Action Owner | Status |
| 001 |  |  |  |
| 002 |  |  |  |
| 003 |  |  |  |
| 004 |  |  |  |
| 005 |  |  |  |
| 006 |  |  |  |
| 007 |  |  |  |

# Approval & Sign Off

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Title | Company | Review Date | Sign Off Date |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |