

Software Requirements Specification (SRS)

Floosy - Personal Finance & Tax Compliance Platform for Oman

Table of Contents

- 1. Introduction**
 - 2. Overall Description**
 - 3. System Features**
 - 4. External Interface Requirements**
 - 5. Non-Functional Requirements**
 - 6. Database Requirements**
 - 7. User Stories**
 - 8. Acceptance Criteria Summary**
-

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document describes the functional and non-functional requirements for Floosy, a simplified web-based personal finance and tax compliance platform designed for individuals and small businesses in Oman. The system helps users track their monthly and annual income and expenses while calculating tax obligations under Omani income tax law.

1.2 Scope

Floosy is a web application that will:

- Allow users to track monthly and annual income from 11 different income sources**
- Enable tracking of 4 types of tax-deductible expenses monthly**
- Calculate annual income tax obligations based on Omani tax laws**
- Generate monthly comparison reports and annual tax summaries**
- Provide a dashboard with financial statistics and insights**

1.3 Definitions and Acronyms

- **Individual User:** A person tracking personal finances and tax obligations
 - **Business User:** A small business owner managing business finances
 - **Tax Year:** The fiscal year for tax calculation purposes in Oman
 - **OMR:** Omani Rial (local currency)
-

2. Overall Description

2.1 Product Perspective

Floosy is a standalone web application built using ASP.NET Core with Angular frontend. The system manages monthly income/expense tracking and annual tax calculations with a focus on simplicity and essential features.

2.2 User Classes

1. Individual User

- Personal income and expense tracking
- Personal tax calculation and planning
- Monthly comparison reports
- Dashboard analytics

2. Business User

- Business income and expense tracking
- Business tax calculation and compliance
- Advanced reporting capabilities
- Multi-category expense management

2.3 Operating Environment

- **Database:** SQL Server
- **Backend:** ASP.NET Core Web API
- **Frontend:** Angular with TypeScript
- **UI Framework:** Bootstrap or Angular Material

- **Browser Support:** Modern web browsers (Chrome, Firefox, Edge, Safari)
-

3. System Features

3.1 User Authentication and Registration

3.1.1 User Registration

Description: New users can register as Individual or Business users

Priority: High

Functional Requirements:

- **Registration form shall include:**
 - **Name**
 - **Email Address (unique)**
 - **Phone Number**
 - **Password (with confirmation)**
 - **User Type Selection (Individual/Business)**
- **System shall validate email uniqueness**
- **System shall require email verification before account activation**
- **System shall enforce password strength requirements**

3.1.2 User Login

Description: Registered users can securely log into the system

Priority: High

Functional Requirements:

- **System shall authenticate users using email and password**
- **System shall provide "Remember Me" functionality**
- **System shall provide "Forgot Password" functionality**
- **System shall redirect users to dashboard after login**

3.2 Income Management

3.2.1 Monthly Income Entry

Description: Users can record monthly income from multiple sources

Priority: High

Functional Requirements:

- **Users shall enter income for specific months with following categories:**
 - **Employment Income**
 - **Self Employment Income**
 - **Rental Income**
 - **Royalty Income**
 - **Interest Income**
 - **Dividend and Sukuk Income**
 - **Real Estate Disposal Gains**
 - **Retirement and EOSB Income**
 - **Prize Income**
 - **Grants and Gifts**
 - **Board Member Compensation**
- **Users can enter amounts for any combination of income types**
- **System shall automatically calculate total monthly income**
- **System shall validate income amounts (positive numbers only)**
- **System shall allow editing of previously entered monthly data**

3.2.2 Annual Income Summary

Description: System provides annual income overview

Priority: Medium

Functional Requirements:

- **System shall calculate total annual income from all monthly entries**
- **System shall display income breakdown by category for the year**
- **System shall show monthly progression throughout the year**

3.3 Expense Management

3.3.1 Monthly Expense Entry

Description: Users can record monthly tax-deductible expenses

Priority: High

Functional Requirements:

- **Users shall enter expenses for specific months with following categories:**
 - **Education Expenses**
 - **Healthcare Expenses**
 - **Zakat**
 - **Interest on Loans/Financing**
- **Users can enter amounts for any combination of expense types**
- **System shall automatically calculate total monthly expenses**
- **System shall validate expense amounts (positive numbers only)**
- **System shall allow editing of previously entered monthly data**

3.3.2 Annual Expense Summary

Description: System provides annual expense overview

Priority: Medium

Functional Requirements:

- **System shall calculate total annual expenses from all monthly entries**
- **System shall display expense breakdown by category for the year**
- **System shall show monthly expense patterns throughout the year**

3.4 Tax Calculation Engine

3.4.1 Annual Tax Calculation

Description: System calculates annual income tax based on Omani law

Priority: High

Functional Requirements:

- **System shall calculate annual tax using Omani tax brackets:**
 - **0% for income up to OMR 30,000**

- Progressive rates for higher income levels (hardcoded in system)
- System shall calculate taxable income (Total Income - Total Deductible Expenses)
- System shall display detailed tax calculation breakdown
- System shall save tax calculations in AnnualCalculation table
- System shall allow recalculation when income/expense data changes

3.5 Dashboard and Analytics

3.5.1 Financial Dashboard

Description: Main dashboard showing key financial metrics

Priority: High

Functional Requirements:

- Dashboard shall display:
 - Current month income and expenses
 - Year-to-date totals
 - Annual tax liability estimate
 - Quick summary cards with key numbers
- Dashboard shall provide navigation to detailed sections
- Dashboard shall show recent activity and updates

3.5.2 Monthly Comparisons

Description: Compare financial performance across months

Priority: Medium

Functional Requirements:

- System shall provide month-to-month comparison charts
- System shall show income and expense trends over time
- System shall display percentage changes between consecutive months
- System shall highlight significant increases or decreases

3.6 Reports

3.6.1 Monthly Reports

Description: Generate monthly financial summaries

Priority: Medium

Functional Requirements:

- **System shall generate monthly reports showing:**
 - **Total income by category**
 - **Total expenses by category**
 - **Net income for the month**
 - **Comparison with previous month**
- **Reports shall be viewable on screen and printable**
- **Reports shall include basic charts and visualizations**

3.6.2 Annual Tax Reports

Description: Generate annual tax summary for filing

Priority: High

Functional Requirements:

- **System shall generate annual tax reports showing:**
 - **Total annual income by category**
 - **Total annual deductible expenses by category**
 - **Taxable income calculation**
 - **Tax liability calculation**
 - **Summary suitable for tax filing**
 - **Reports shall be formatted for easy reference during tax filing**
-

4. External Interface Requirements

4.1 User Interface Requirements

- **Responsive Design:** Interface shall work on desktop, tablet, and mobile devices

- **Modern Framework:** UI shall use Bootstrap or Angular Material for consistent styling
- **Navigation:** Clear and intuitive navigation structure
- **Language:** Interface shall be in English
- **Currency:** All amounts displayed in Omani Rial (OMR) with proper formatting

4.2 Software Interfaces

- **Database:** SQL Server integration with Entity Framework
 - **Authentication:** ASP.NET Core Identity for user management
-

5. Non-Functional Requirements

5.1 Performance Requirements

- **Response Time:** Pages shall load within 2 seconds under normal load
- **Concurrent Users:** System shall support up to 500 concurrent users
- **Database Performance:** Tax calculations shall complete within 5 seconds
- **Report Generation:** Monthly reports shall generate within 30 seconds

5.2 Security Requirements

- **Data Encryption:** All financial data shall be encrypted at rest and in transit
- **Password Security:** Passwords shall be hashed using bcrypt with salt
- **Session Management:** Secure session handling with 30-minute timeout
- **Input Validation:** All user inputs shall be validated and sanitized
- **HTTPS:** All communication shall use HTTPS encryption

5.3 Reliability Requirements

- **Uptime:** System shall have 99.5% uptime
- **Data Backup:** Automated daily backups
- **Error Handling:** Graceful error handling with user-friendly messages
- **Data Integrity:** Financial data shall maintain consistency and accuracy

5.4 Usability Requirements

- **User Experience:** Intuitive interface requiring minimal training
 - **Error Messages:** Clear and helpful error messages
 - **Mobile Experience:** Fully functional mobile interface
-

6. Database Requirements

6.1 Core Tables Structure

User Table

- **ID (Primary Key) - INT IDENTITY**
- **Name - NVARCHAR(100)**
- **Email - NVARCHAR(150) UNIQUE**
- **Password - NVARCHAR(255)**
- **PhoneNumber - NVARCHAR(20)**
- **UserType - NVARCHAR(20) DEFAULT 'Individual'**
- **CreateAt - DATETIME2 DEFAULT GETDATE()**

Income Table

- **ID (Primary Key) - INT IDENTITY**
- **EmploymentIncome - DECIMAL(18,3) DEFAULT 0**
- **SelfEmploymentIncome - DECIMAL(18,3) DEFAULT 0**
- **RentalIncome - DECIMAL(18,3) DEFAULT 0**
- **RoyaltyIncome - DECIMAL(18,3) DEFAULT 0**
- **InterestIncome - DECIMAL(18,3) DEFAULT 0**
- **DividendAndSukukIncome - DECIMAL(18,3) DEFAULT 0**
- **RealEstateDisposalGains - DECIMAL(18,3) DEFAULT 0**
- **RetirementAndEosbIncome - DECIMAL(18,3) DEFAULT 0**
- **PrizeIncome - DECIMAL(18,3) DEFAULT 0**

- GrantsAndGifts - DECIMAL(18,3) DEFAULT 0
- BoardMemberCompensation - DECIMAL(18,3) DEFAULT 0
- UserID (Foreign Key to User) - INT
- MonthYear - DATE
- CreateAt - DATETIME2 DEFAULT GETDATE()

Expenses Table

- ID (Primary Key) - INT IDENTITY
- EducationExpenses - DECIMAL(18,3) DEFAULT 0
- HealthcareExpenses - DECIMAL(18,3) DEFAULT 0
- Zakat - DECIMAL(18,3) DEFAULT 0
- InterestOnLoansFinancing - DECIMAL(18,3) DEFAULT 0
- UserID (Foreign Key to User) - INT
- MonthYear - DATE
- CreateAt - DATETIME2 DEFAULT GETDATE()

AnnualCalculation Table

- ID (Primary Key) - INT IDENTITY
- Year - INT
- TotalIncome - DECIMAL(18,3) DEFAULT 0
- TotalExpenses - DECIMAL(18,3) DEFAULT 0
- TaxableIncome - DECIMAL(18,3) DEFAULT 0
- Tax - DECIMAL(18,3) DEFAULT 0
- UserID (Foreign Key to User) - INT
- CreateAt - DATETIME2 DEFAULT GETDATE()

7. User Stories

7.1 Individual User Stories

1. As an Individual User, I want to enter my monthly income from different sources so that I can track my earnings over time
2. As an Individual User, I want to record my monthly tax-deductible expenses so that I can maximize my tax savings
3. As an Individual User, I want to see my annual tax calculation so that I know how much tax I need to pay
4. As an Individual User, I want to view my dashboard with key financial metrics so that I can quickly understand my financial status

API Endpoints

Authentication

POST /api/auth/register - User registration

POST /api/auth/login - User login

POST /api/auth/logout - User logout

POST /api/auth/forgot - Password reset

Income Management

GET /api/income/{userId}/{year} - Get annual income

GET /api/income/{userId}/{year}/{month} - Get monthly income

POST /api/income - Create/Update monthly income

DELETE /api/income/{id} - Delete income record

Expense Management

GET /api/expenses/{userId}/{year} - Get annual expenses

GET /api/expenses/{userId}/{year}/{month} - Get monthly expenses

POST /api/expenses - Create/Update monthly expenses

DELETE /api/expenses/{id} - Delete expense record

Tax Calculation

GET /api/tax/calculate/{userId}/{year} - Calculate annual tax

GET /api/tax/history/{userId} - Get tax calculation history

POST /api/tax/recalculate/{userId}/{year} - Recalculate taxes

Reports & Analytics

GET /api/reports/monthly/{userId}/{year}/{month} - Monthly report

GET /api/reports/annual/{userId}/{year} - Annual tax report

GET /api/dashboard/{userId} - Dashboard data

GET /api/analytics/comparison/{userId}/{year} - Monthly comparisons