**Cura Senior Backend Software Engineer Challenge**

**Designing an Email System**

## 1.1 Introduction

The main purpose of this assignment is to let you show off your personal development skills and ability that help you to focus to deliver clean, organized, high-quality and easy to understand code and you can use any modern technologies and any readymade or 3rd party plugins and controls if you like.

You are tasked with building an Email System that consists of the following pages:

* Welcome page
* Registration page
* Inbox page
* Sent Items page
* Contacts List page

## Guidelines:

1. Please feel free to any modern technologies and any readymade or 3rd party plugins and controls if you need to.
2. If you have technical questions and feel stuck about your task, just make an assumption and write a comment to clarify your assumption and proceed.
3. You can use the following technologies to apply for this assignment:
   * ASP.NET MVC Framework C# or ASP.NET Web Forms C#
   * JQuery (is a must)
   * CSS, CSS3
   * Entity Framework, NHibernate or ADO.NET
   * MS SQL
   * Any readymade or 3rd party plugins & controls.

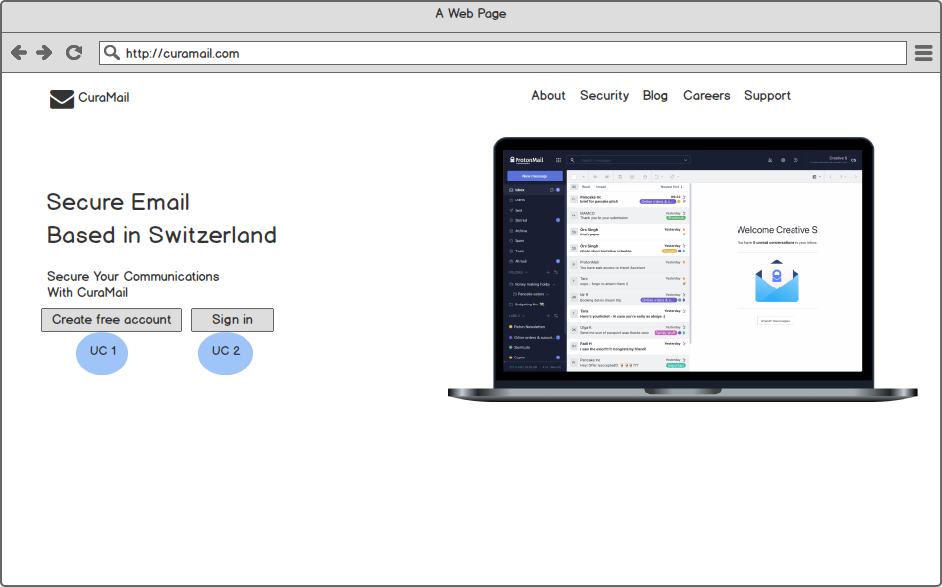
Task Deliverables:

1. A reply to the original email you received containing the below :
   1. A URL to your web solution hosted on on any free hosting provider so we can see your pages live.
   2. A URL to a GitHub or Bitbucket repo or any other source control services containing the source code of your solution.
2. If you want to clarify something or explain other stuff, feel free to write that in your reply email.

## 1.2 Welcome Screen

### 1.2.1 System Analysis

#### 1.2.1.1 View Welcome Screen



Above is an image showing a suggested design for welcome screen, the main idea is to focus on the main user actions (sign in and create a free account). Do not be restricted in this design if you have a suggestion for a better design then do it keeping in mind the simplicity of the design.

**UC1**: This use case starts when a user wishes to log into the CuraMail System

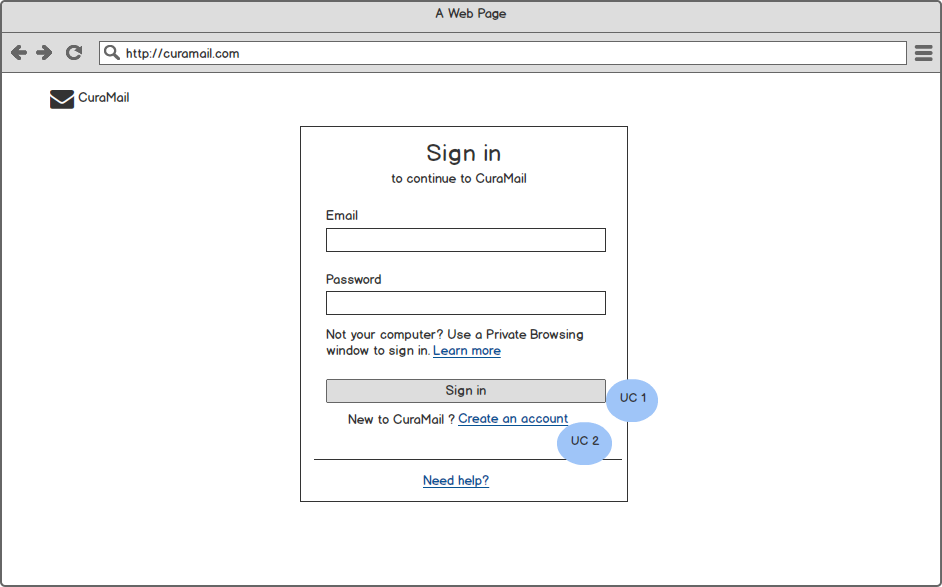
**UC2**: User can navigate to create a free account screen.

### 1.2.2 Business Analysis

## 1.3 Sign in Screen

### 1.3.1 System Analysis

#### 1.3.1.1 View Sign in Screen



The image above is showing a user logs into the CuraMail System.

**UC1**: The system requests that the user enter his/her email and password and will validates the entered email and password and logs the user into the system, [UC: Sign In](#_1.3.2.2_UC:_Sign)

**Alternative Flows** (Invalid Name / Password)

If in the Basic Flow the user enters an invalid email and/or password, the system displays an error message.

**UC2**: User can navigate to create a free account screen, [UC: Create a free account](#_1.4.2.1_UC:_Registration)

### 1.3.2 Business Analysis

#### 1.3.2.2 UC: Sign in

**Authorization**: No Authorization

**Paths**:

1. The user will go to Welcome screen >> Sign in
2. The user can access Sign in screen directly through the URL.

**Dependency**: No Dependency.

**Business Requirement**: If the sign in process was successful, the user is now logged into the system. If not, the system state is unchanged

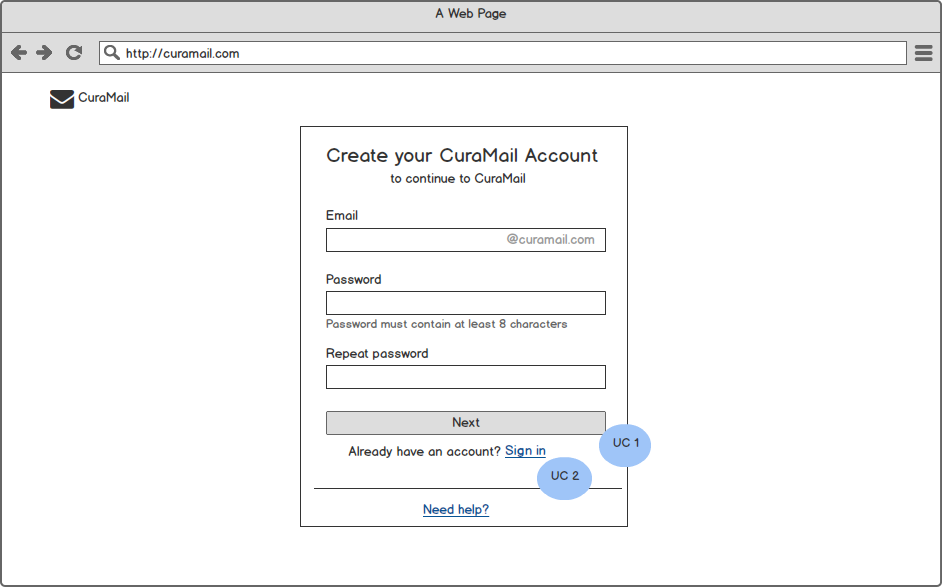
**Sections**: The sign in screen will contain the following sections:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Mandatory** | **Description** | **Visibility** |
| Email | Required | User Email | Always Visible |
| Password | Required | Password length of at least 8 characters | Always Visible |

## 1.4 Registration Screen

### 1.4.1 System Analysis

#### 1.4.1.1 View Registration Screen 1



The image above is showing the first step of registering new users. The validation can be implemented server side or client side or both, do it in way that ensures the security.

*“Needs help”* link is not required to navigate to any page.

**UC1**:

1. User types a valid email of his or her choice  
2. User types a password  
3. User retypes the password  
Next  
4. System checks if the user email is not already in use  
5. System checks if the two passwords are identical  
6. System registers the new player with the given parameters (user email, password)

**Alternative Flows** (User email is already in use)

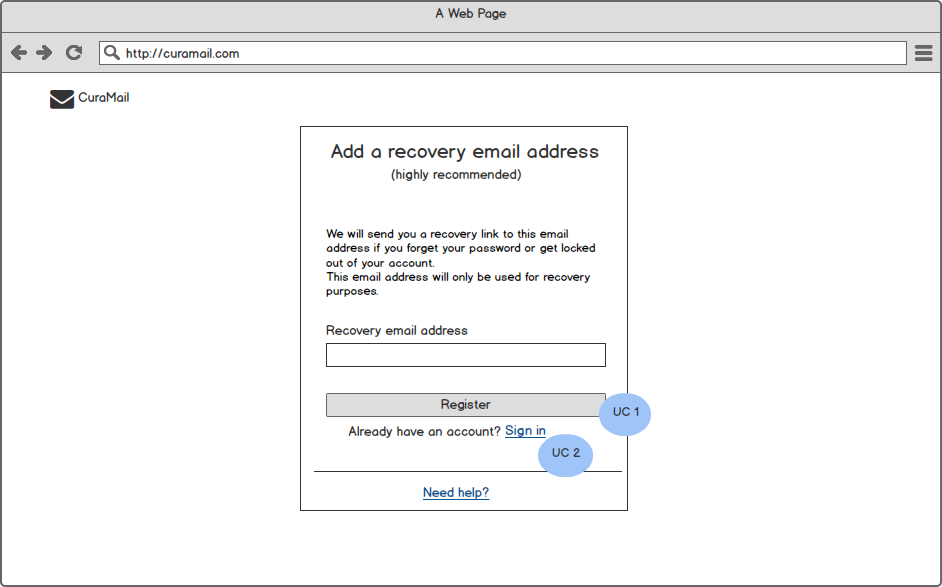
User is requested to select another email.

**Alternative Flows** (The two passwords are different)

User is requested to retype (twice) his/her password

**UC2**: User can navigate to Sign in screen, [UC: Sign In](#_1.3.2.2_UC:_Sign)

#### 1.4.1.2 View Registration Screen 2



The image above is showing the second step of registering new users.

**UC1:** the user can set a recovery email and the System checks if the email as it should be different from the user's email and this step is optional and the user can skip it.

**UC2**: User can navigate to Sign in screen, [UC: Sign in](#_1.3.2.2_UC:_Sign)

### 1.4.2 Business Analysis

#### 1.4.2.1 UC: Registration Screen step one

**Authorization**: No Authorization

**Paths**:

1. The user will go to Welcome screen >> Create a free account.
2. The user can access registration screen directly through the url.

**Dependency**: No Dependency.

**Business Requirement**: If the registration process was successful, the user becomes a registered player and logged into the system and the system will navigate the user to the second step of the registration process. If not, the system state is unchanged

**Sections**: The Registration screen will contain the following sections:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Mandatory** | **Description** | **Visibility** |
| Email | Required | User Email | Always Visible |
| Password | Required | Password length of at least 8 characters | Always Visible |
| Repeat password | Required | User is requested to retype his/her password | Always Visible |

#### 1.4.2.2 UC: Recovery Email Screen

**Authorization**: any authenticated user will be authorized to see this screen.

**Paths**: The user will go to registration step one >> add recovery email address.

**Dependency**: The user should be authenticated.

**Business Requirement**: This step is optional and the user can skip it but if he adds a recovery mail then the system should send a notification to the recovery email and if the send status is true then the system will mark this email as verified.

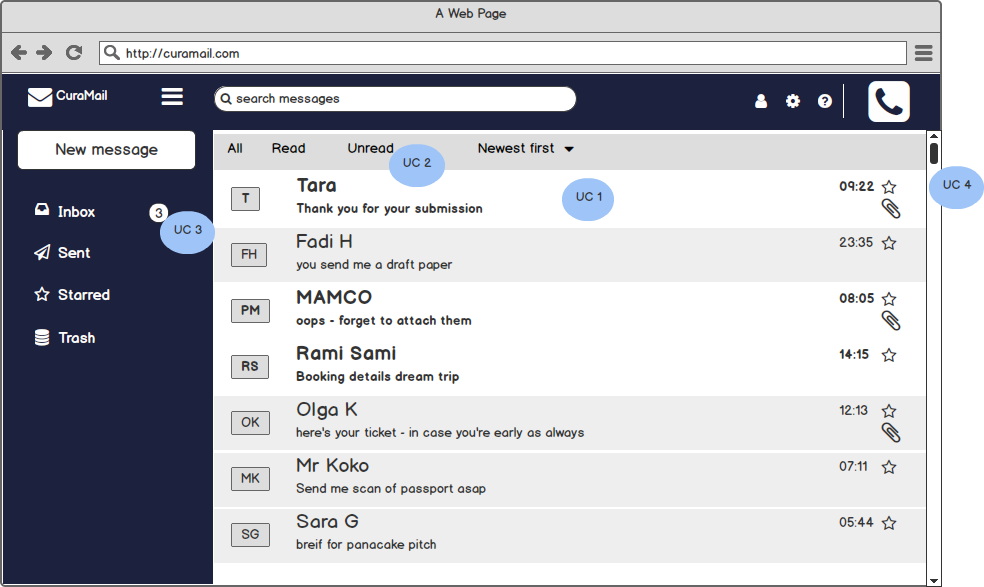
**Sections**: The Recovery email screen will contain the following sections:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Mandatory** | **Description** | **Visibility** |
| Email | Optional | Recovery email address | Always Visible |

## 1.5 CuraMail Dashboard

### 1.5.1 System Analysis

#### 1.5.1.1 View Dashboard Screen



The image above is showing CuraMail Dashboard where the inbox folder is selected by default and the user can navigate between the other email folders.Starred email action in the emails list is not required to implemented.

**UC1**: the system will display the new and old emails and he will be able to see the attachment flag if the email contains attachment and the most important emails “Starred emails” ordered by date and time.

**UC2**: the user can filter the displayed emails to see read, unread and all mails in the selected folder always newest first should be selected.

**UC3**: the system will display the number of new emails in each folder.

**UC4:** the system will display the old emails for the selected folder when the user scrolls down

### 1.5.2 Business Analysis

**Authorization**: any authenticated user will be authorized to see this screen

**Paths**:

1. The user will go to Welcome screen >> sign in >> Dashboard.
2. The user will go to Welcome screen >> Create a free account >> Dashboard.

**Dependency**: the user should be authenticated.

**Business Requirement**: No Business requirement