# 

# **Data Definitions**

Table 1 - Table of Data Definitions

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Meaning** | **Usage** | **Comment** |
| Registered User | Actor | *Use Case Scenarios* | A student who has registered with the system |
| Unregistered User | Actor | *Use Case Scenarios* | A student who has not registered with the system |
| System | Platform Hardware and Services | *Use Case Scenarios* | The collective intercommunicating software and hardware components forming a computer system, including front-end and back-end code, configuration files, documentation, and Firebase database |
| Web Site | User Interface | *User Interface* | Front-end display for user interaction |
| Home Page | User Interface | *User Interface* | First web page the user sees when not logged in |
| Navigation Bar | Service/User Interface | *Site User Service* | Fixed toolbar display containing a variety of site navigation pathways (Home, Support, Settings, Add Course, Log Out) |
| Dashboard | Service/User Interface | *Site User Service* | Primary hub for displaying student information and tools |
| Account | Data | *Use Case Scenarios* | Refers to the location on a network server storing the user ID, password, and composite information associated with an individual user |
| Login | Service | *Site User Service* | Permits user to access their stored personal account data and grade analysis tools |
| Session | Service | *Site User Service* | Period of activity between a user logging in and logging out of a multi-user system |
| Logout | Service | *Site User Service* | Ends a user’s login session and redirects to home page |
| Register | Service | *Site User Service* | Process by which an unregistered user may become a registered user |
| User ID | Data | *Use Case Scenarios* | A unique logical entity and name of respective user’s email address used to identify and distinguish between user |
| Password | Data | *Use Case Scenarios* | A confidential, user-defined string of characters used to authorize access to the account of an associated UserID |
| Forgot Password | Service | *Site User Service* | A self-service account recovery process by which a user may choose to create a new password after clicking the password reset link sent to the associated user’s email address |
| Password Reset Link | Off-site Service | *Off-site User Service* | An automated link sent to a user’s registered email address that contains a unique token identifying the user which is authenticated by second factor when the link is clicked |
| Confirmation Email | Off-site Service | *Off-site User Service* | An automated email triggered upon registration of a new user which must be addressed within 24 hours to verify user’s identity and initiate a permanent account |
| Account Settings | Service | *Site User Service* | The configurations and settings associated with an authorized account which are established upon initial enrollment, but may also be changed at a later date |
| Course Form | Data | *Use Case Scenarios* | A formatted document containing blank fields in which users must input corresponding data necessary for site function, which is then stored in the database |
| Support | Service | *Site User Service* | Means by which a user may report bugs and issues experienced during website operation to its creators |
| GitHub | Repository Hosting | *Use Case Scenarios* | An internet hosting service for software development and version control using Git whereupon the project code repository and documentation are stored and organized |
| GitHub Pages | Production Server | *Use Case Scenarios* | The website hosting service on which this project is published |
| Firebase | Database/Back-End Framework | *Use Case Scenarios* | A backend-as-a-service (Baas) which provides real-time database hosting and user authentication services |

# **List of High-Level Functional Requirements (Initial)**

## Unregistered User

1. **Account Creation**
   1. **Procedural Steps**
2. User selects ‘Register’ button from the homepage navigation bar
3. User enters a new UserID (same as valid email address)
4. User enters a Password:
   1. System will check if password is *at least* six characters in length:
5. Meets Criteria – Borders of password input field box returns to default settings; Go to Step IV
6. Fails Criteria – Borders of password input field box turn red and a notification appears informing the user that passwords must be *at least* six characters in length; Go to Step III
7. User re-enters Password:
   1. System will check if both password fields match:
      1. Passwords Match – Borders of password input field boxes return to default settings; Go to Step V
      2. Passwords Do Not Match – Borders of password input field boxes turn red and a notification appears informing the user of non-matching passwords; Go to Step III
8. User confirms via ‘Submit’ selection
9. System will check if UserID is available:
   1. Available –
      1. System will send email prompting confirmation to user.
      2. A temporary account will be created in Firebase
      3. User will be notified to address confirmation email within 24 hours to prevent account deletion; Go to Step VII
   2. Unavailable – borders of UserID input field box turn red and a notification appears informing the user of UserID unavailability and to try another UserID; Go to Step II
10. User may choose to confirm account validity via primary email:
    1. User Confirms –
       1. User clicks confirmation link within 24 hours
       2. System permanently saves UserID and password 🡪 End of process
    2. User Does Not Confirm –
       1. User does not click confirmation link within 24 hours
       2. System will delete account information at 24 hour mark 🡪 End of process

## Registered User

1. **Account Login**
   1. **Login Procedure**
2. User selects ‘Login’ button from the homepage navigation bar
3. User enters UserID
4. User enters Password associated with UserID
5. User selects ‘Login’ button:
6. System will check if the entered UserID & Password pairing matches:
   1. Valid Login Combination – System shall redirect user to the dashboard 🡪 End of process
   2. Invalid Login Combination – System shall alert user that current UserID & Password combination is not found in the database; Go to Step II
   3. **Password Recovery**
7. User selects ‘Forgot Password?’ button beneath the ‘Login’ button
8. System will redirect user to ‘Password Reset’ page
9. User enters UserID into the Email Address field
10. User selects ‘Submit’ button
11. System will notify user that “a password reset link has been sent to the provided email address if the UserID exists within the system”
12. System will check if UserID exists in the system:
13. UserID Exists – System shall send a password reset link to the provided email address; Go to Step VII
14. UserID Does Not Exist – System shall take no further action 🡪 End of process
15. User clicks password reset link from their primary email address
16. System will redirect user to ‘New Password’ page
17. User enters new Password:
18. System will check if password is *at least* six characters in length:
19. Meets Criteria – Borders of password input field box returns to default settings; Go to Step X
20. Fails Criteria – Borders of password input field box turn red and a notification appears informing the user that passwords must be *at least* six characters in length; Go to Step IX
21. User re-enters new Password:
22. System will check if both password fields match:
23. Passwords Match – Borders of password input field boxes return to default settings; Go to Step XI
24. Passwords Do Not Match – Borders of password input field boxes turn red and a notification appears informing the user of non-matching passwords; Go to Step IX
25. System redirects user to the application homepage to login with new password 🡪 End of process
26. **Data Entry**
    1. **Account Settings**
27. User selects ‘Account Settings’ button (cogwheel icon) from the navigation bar
28. User enters name of school
29. User selects current semester and year
30. User confirms existing GPA scale or enters new parameters
31. User may opt to enter phone number for text notifications
32. User clicks ‘Submit’ 🡪 End of process
    1. **Course Form**
       1. **Course Name**
33. User inputs a three-letter acronym
    1. On input system will automatically capitalize and limit input length to 3 characters
       1. **Term Length**
34. User clicks on field
35. Dropdown appears
36. User may choose from list of pre-designated options
    * 1. **Credit Hours**
37. User clicks on field
38. Dropdown appears
39. User may choose from list of pre-designated options
    * 1. **Class Days**
40. User selects either day(s) of the week or remote checkbox option
41. System will check if remote option is selected:
42. Selected – unchecks and prevents further interaction with options involving days of the week
43. Unselected – allows user to select day(s) of the week
    * 1. **Grading Scale**
44. User selects either from two pre-made grading scales or a custom grading scale
45. System will check if custom option is selected:
46. Selected – custom grading scale input fields appear
47. Unselected – custom grading scale input fields disappear
    * 1. **Weighted Categories**
48. User selects the categories of weighted groupings
49. System will check if any options are selected:
50. For EACH Selected – two fields appear:
    * 1. Quantity – User inputs the numerical total of items corresponding to category, and system will check:
         1. < 1 – notifies user numerical value must be greater than or equal to 1
         2. > 1 – nothing
      2. Percentage – User inputs the numerical percentage of total grade corresponding to category, and system will check:
         1. < 1 or > 100 – notifies user numerical value must be between 1 and 100
         2. > 1 and < 100 – nothing
51. For EACH Unselected – two fields disappear
    * 1. **Form Submission**
52. User clicks ‘Submit’ and input validation is executed
53. System will check if ‘Custom’ grading scale custom is selected:
54. True – System will check grading bounds for each field in hierarchal order from A-F for numerical legitimacy:
55. If ANY Above < Below, > 100, or < 1 – Notify user of invalid parameters; Go to Step 3.2.5
56. If ALL Above > Below and between 1 and 100 – Condition Passed; Go to Step B
57. False – Go to Step B
58. System will sum Weighted Category percentage fields and evaluate if total equals 100:
59. True – Go to Step II
60. False – Notify user that total percentages must equal 100; Go to Step 3.2.6
61. User data is successfully stored in the database 🡪 End of process
    1. **Course Dates**
62. The course dates form immediately follows submission of Course Form
63. A new date field is generated entitled “<Category Name> #<Number>:” for each individual course deliverable based upon the item quantity designated for each weighted category on the previous Course Form
64. User interacts with the calendar API to assign the deadline for each item
65. User clicks ‘Submit’ and data is successfully stored 🡪 End of process
    1. **Grade Entry**
66. User selects ‘Grade Entry’ button (report card icon) from the dashboard
67. A new grade field is generated entitled “<Category Name> #<Number>:” for each deliverable item with a deadline prior to the current date
68. For each deliverable item with known scoring feedback, the user may input the corresponding percentage value between 1 and 100 up to four decimal places
69. User clicks ‘Submit’ and data is successfully stored 🡪 End of process