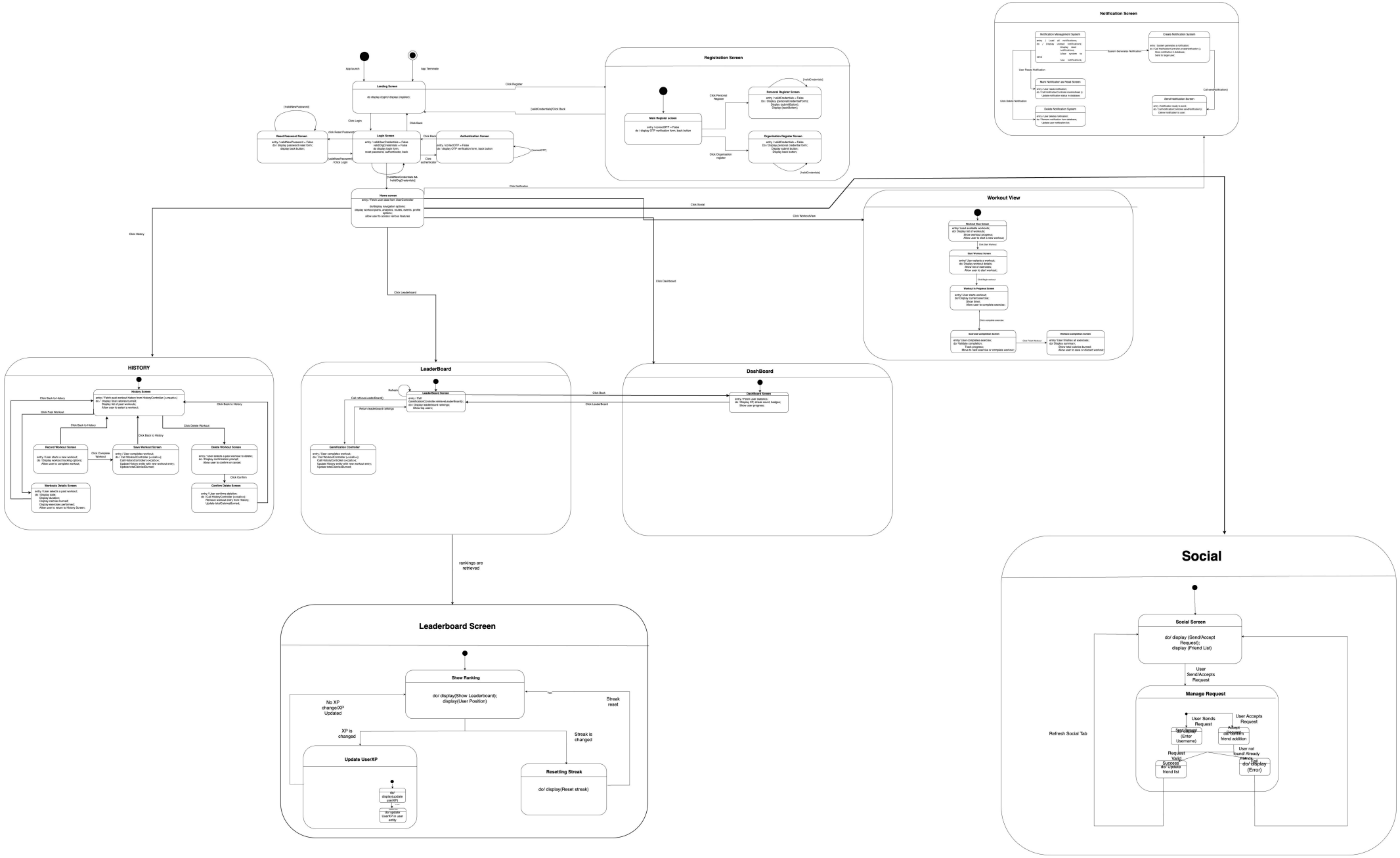
App launch
App Terminate
Landing Screen
do display (login/ display (register);
Click Register
[validCredentials]/Click Back
Click Login
Click Back
Reset Password Screen
entry / validNewPassword = False
do / display password reset form;
display back button;
[validNewPassword]
click Reset Password
Login Screen
entry / validUserCredentials = False
validOtpCredentials = False
do display login form,
reset password, authenticate, back
[validNewPassword] / Click Login
Click Back
Authentication Screen
entry / correctOTP = False
do / display OTP verification form, back button
[correctOTP]
Click authenticate
[validNewCredentials && validOtpCredentials]
Registration Screen
Main Register screen
entry / correctOTP = False
do / display OTP verification form, back button
Click Personal Register
Click Organisation register
Personal Register Screen
entry / validCredentials = False
Do / Display (personalCredentialForm);
Display submit button;
Display (backButton);
[validCredentials]
Organization Register Screen
entry / validCredentials = False
Do / Display personal credential form;
Display submit button;
Display back button;
[validCredentials]
Click Notification
Home screen
entry / Fetch user data from UserController
do/display navigation options;
display workout plans, analytics, routes, events, profile options;
allow user to access various features
Click Social
Click WorkoutView
Click History
Click Leaderboard
Click Dashboard
Notification Screen
Notification Management System
entry / Load all notifications;
do / Display unread notifications;
Display read notifications;
send Notification to server.
User Reads Notification
Click Delete Notification
Mark Notification as Read Screen
entry / User reads notification;
do / Call NotificationController.markAsRead ();
Update notification status in database.
Delete Notification System
entry / User deletes notification;
do / Remove notification from database;
Update user notification list.
System Generates Notification
Create Notification System
entry / Screen generates a notification;
do / Call notificationCreate (userId,notification);
Store notification in database;
Send to origin user.
Call sendNotification()
Send Notification Screen
entry / Notification ready to send;
do / Call notificationController.sendNotification();
Deliver notification to user;
Workout View
Workout View Screen
entry/ Load available workouts;
do/ Display list of workouts;
Show workout progress;
Allow user to start a new workout.
Click Start Workout
Start Workout Screen
entry/ User selects a workout;
do/ Display workout details;
Show list of exercises;
Allow user to start workout;
Click begin workout
Workout In Progress Screen
entry/ User starts workout;
do/ Display current exercise;
Show timer;
Allow user to complete exercise;
Click complete exercise
Exercise Completion Screen
entry/ User completes exercise;
do/ Mark exercise as complete;
Track progress;
Move to next exercise or complete workout;
Click Finish Workout
Workout Completion Screen
entry/ User finishes all exercises;
do/ Display summary;
Show total calories burned;
Allow user to save or discard workout.
HISTORY
History Screen
entry / Fetch past workout history from HistoryController (uniqueId=);
do / Display total calories burned;
Display list of past workouts;
Allow user to select a workout;
Click Back to History
Click Past Workout
Click Delete Workout
Record Workout Screen
entry / User starts a new workout;
do / Display workout tracking options;
Allow user to complete exercise;
Click Complete Workout
Save Workout Screen
entry / User completes workout;
do / Call WorkoutController (uniqueId=);
Call HistoryController (uniqueId=);
Update HistoryLastWeekLoad with new workout entry;
Update totalCaloriesBurned;
Delete Workout Screen
entry / User selects a past workout to delete;
do / Display confirmation prompt;
Allow user to confirm or cancel;
Click Confirm
Workouts Details Screen
entry / User selects a past workout;
do / Display date;
Display duration;
Display calories burned;
Display exercises performed;
Allow user to return to History Screen;
Confirm Delete Screen
entry / User confirms deletion;
do / Call HistoryController (uniqueId=);
Remove workout from History;
Update totalCaloriesBurned;
LeaderBoard
Refresh
LeaderBoard Screen
entry / Call GamificationController.retrieveLeaderBoard();
do / Display leaderboard rankings;
Show fail users;
Call retrieveLeaderBoard()
Return leaderboard rankings
Gamification Controller
entry / User completes workout;
do / Call WorkoutController (uniqueId=);
Call HistoryController (uniqueId=);
Update HistoryLastWeekLoad with new workout entry;
Update totalCaloriesBurned;
Click Back
Click LeaderBoard
DashBoard
DashBoard Screen
entry / Fetch user statistics;
do / Display XP, streak count, badges;
Show user progress.
rankings are retrieved
Leaderboard Screen
Show Ranking
do/ display(Show Leaderboard);
display(User Position)
No XP change/XP Updated
XP is changed
Streak reset
Streak is changed
Update UserXP
do/ display(update userXP)
do/ update UserXP in user entity
Resetting Streak
do/ display(Reset streak)
Social
Social Screen
do/ display (Send/Accept Request);
display (Friend List)
User Send/Accepts Request
Manage Request
User Sends Request
User Accepts Request
do/ display (Enter Username)
Accept request
do/ display friend addition
Request Valid
User not found/ Already friend
Success
do/ update friend list
do/ display (Error)
Refresh Social Tab