

Element	Detail
Name	Create Account
Description	This use case describes how a new user creates an account on Lunki by providing their email, username, and password
Actor(s)	User
Triggers	A user decides they would like to create a Lunki account
Preconditions	<ol style="list-style-type: none"> 1. Email address or username provided are not attached to an existing account 2. Email address provided is valid
Postconditions	<ol style="list-style-type: none"> 1. A new account is created and stored in the system
Main Success Path	User creates an account using their email, username, and password. It is then stored in the system.
Actor Actions	System Response
1. User opens Lunki	2. System displays login/create account options
3. User selects "create account"	4. System displays email address, username, and password fields
5. User completes information fields	6. System validates inputs: <ul style="list-style-type: none"> - Validates that email or username are not already in use - Validates password strength - Validates email address is valid 7. System creates a new account and stores it in database
Exception Path E1-7 a	User inputs invalid information
Actor Actions	System Response
	7. System detects invalid sign up information
	8. System displays create account error message, prompts user to input correct information
	9. Use Case ends

Element	Detail
Name	Save Workout
Description	This use case describes how a user can save a workout tutorial from the workout database
Actor(s)	User
Triggers	A user decides they would like to save a workout tutorial
Preconditions	1. User is logged into Lunki
Postconditions	1. A workout tutorial is added to the User's save folder
Main Success Path	User views a workout from the database, saves it, and the workout tutorial is added to the User's save folder
Actor Actions	System Response
1. User selects a workout tutorial	2. System displays workout tutorial information and options
3. User saves the workout	4. System adds the workout tutorial to the User's save folder

Element	Detail
Name	Add Friend
Description	This use case describes how a user can add a friend by inputting their username
Actor(s)	User
Triggers	A user decides they would like to add a friend
Preconditions	1. User is logged into Lunki
Postconditions	1. A friend request is sent to the other user
Main Success Path	User inputs the username of another user they'd like to friend. A request is then sent to that user.
Actor Actions	System Response
1. User selects add friend	2. System displays username field
3. User enters username of friend	4. System validates username is valid 5. System sends friend request to the user with that username
Alternative Path A1-1a	User adds friend from mutual groupchat
Actor Actions	System Response
1. User selects group chat	2. System displays groupchat members
3. User selects the profile of another user to friend	4. System displays other user's profile
5. User selects "send request"	6. System sends friend request to the other user
Exception Path E1-5 a	User inputs invalid username
Actor Actions	System Response
	5. System detects invalid username
	6. System displays invalid username error message, prompts user to input valid username
	7. Use Case ends

Element	Detail
Name	Join Groupchat
Description	This use case describes how a user can join a groupchat using an invite code
Actor(s)	User
Triggers	A user receives an invite code to a group chat
Preconditions	<ol style="list-style-type: none"> 1. User is logged into Lunki 2. User has a valid invite code
Postconditions	<ol style="list-style-type: none"> 1. User is added to a groupchat
Main Success Path	User inputs the invite code they've received to join a group chat. They are then added to the chat.
Actor Actions	System Response
1. User selects join group chat	2. System displays invite code field
3. User enters invite code	<ol style="list-style-type: none"> 4. System validates invite code is valid 5. System adds user to the group chat
Exception Path E1-5 a	User inputs invalid invite code
Actor Actions	System Response
	5. System detects invalid invite code
	6. System displays invalid invite code error message, prompts user to input valid code
	7. Use Case ends

Element	Detail
Name	Start Workout
Description	This use case describes how a user can start a workout
Actor(s)	User
Triggers	A user decides they want to start a workout
Preconditions	<ol style="list-style-type: none"> 1. User is logged into Lunki 2. User does not have an ongoing workout
Postconditions	<ol style="list-style-type: none"> 1. A workout is started
Main Success Path	User starts a workout
Actor Actions	System Response
1. User selects start workout	<ol style="list-style-type: none"> 2. System validates a workout is not already in progress 3. System begins a workout, logs date and time of start.
Exception Path E1-3 a	Workout is already in progress
Actor Actions	System Response
	3. System detects user has an active workout
	4. System displays error message, prompts user to end current workout before starting a new one
	5. Use Case ends

Element	Detail
Name	End Workout
Description	This use case describes how a user can end a workout
Actor(s)	User
Triggers	A user decides they want to finish their workout
Preconditions	<ol style="list-style-type: none"> 1. User is logged into Lunki 2. User has an ongoing workout
Postconditions	<ol style="list-style-type: none"> 1. A workout is ended 2. Workout information is logged in the User's workout history
Main Success Path	User finishes a workout. Time elapsed and vitals are stored in user's workout history
Actor Actions	System Response
1. User selects end workout	<ol style="list-style-type: none"> 2. System validates there is an active workout 3. System ends workout, logs date and time of end
	4. System stores time elapsed and fetches vitals during that time (if applicable)
Exception Path E1-3 a	No active workout
Actor Actions	System Response
	3. System detects user does not have an active workout
	4. System displays error message, prompts user to start a workout
	5. Use Case ends

Element	Detail
Name	Set Goal
Description	This use case describes how a user can set themselves a goal
Actor(s)	User
Triggers	A user has a goal they would like to set in Lunki
Preconditions	1. User is logged into Lunki
Postconditions	1. A goal is created with the given information
Main Success Path	User sets a workout goal with a description and timeframe.
Actor Actions	System Response
1. User selects set goal	2. System displays description field
3. User enters a description for their goal	4. System stores description 5. System displays date field
6. User enters start and end dates for their goal	7. System validates dates are valid 8. System creates a goal with specified description and timeframe, stores in User's goal list.
Exception Path E1-8 a	Invalid start/end date
Actor Actions	System Response
	8. System detects invalid dates (end date before beginning date, date has already passed, etc.)
	9. System displays error message, prompts user to enter valid dates
	10. Use Case ends

Element	Detail
Name	Post Workout
Description	This use case describes how a user can post a workout they have completed
Actor(s)	User
Triggers	A user completes a workout and would like to post about it
Preconditions	<ol style="list-style-type: none"> 1. User is logged into Lunki 2. User has at least 1 friend
Postconditions	<ol style="list-style-type: none"> 1. A post is created of the User's workout
Main Success Path	User posts a workout with a description and image, added to friend's feeds.
Actor Actions	System Response
1. User selects create post	2. System displays workout description field
3. User enters a description of the workout they have completed	<ol style="list-style-type: none"> 4. System stores description 5. System displays image field
6. User submits an image of themselves completing the workout	7. System validates image follows terms of service
8. User confirms workout post	<ol style="list-style-type: none"> 9. System creates a post with the given description and image 10. System adds post to the feed of all the User's friends.
Exception Path E1-8 a	Image does not follow terms of service
Actor Actions	System Response
	8. System detects inappropriate or offensive image
	9. System displays inappropriate content message, prompts user to enter an appropriate image
	10. Use Case ends