

User Stories

Projekt AskHsWorms

Fragen App für Studierende und Mitarbeiter:innen der Hochschule Worms

Core - Extension

USER ROLLEN: STUDENT, DOZENT, (TUTOR,) ADMIN

Ich als ... will .. , damit ...

Student:

Ich als Student will mich registrieren können, damit ich ein Konto einlegen kann.

Ich als Student will mich einloggen können, damit ich Fragen stellen kann.

Ich als Student will Neuigkeiten von der Uni sehen, damit ich auf dem Laufenden bleiben.

Ich als Student will von der App direkt Zugriff auf Moodle haben, damit ich schnell zu dem Kurs komme.

Ich als Student will von der App direkt Zugriff auf LSF haben, damit ich schnell den Studienplan sehen kann.

Ich als Student will Fragen stellen, damit ich Antworten bekomme.

Ich als Student will Fragen von anderen sehen und beantworten, damit ich anderen helfen kann.

Ich als Student will Fragen/Beiträgen bewerten, damit ich meine Meinung äußern kann.

Ich als Student will meine Frage löschen, damit es keiner mehr sehen kann.

Ich als Student will meine Frage modifizieren, damit ich es richtig umformuliere.

Ich als Student will Notizen für mich einlegen, damit ich es nicht vergesse.

Ich als Student will Notizen löschen, damit ich sie nicht sehe.

Ich als Student will Notizen bearbeiten, damit sie nicht falsch bleiben.

Dozent:

Ich als Dozent will einen Beitrag (Neuigkeiten) stellen, damit alle Studierenden das sehen können.

Ich als Dozent will Fragen beantworten, damit Studierenden helfen kann.

Ich als Dozent will mich einloggen können, damit ich Beiträge/Fragen stellen kann.

Ich als Dozent will mich registrieren können, damit ich ein Konto einlegen kann.

Ich als Dozent will Notizen für mich einlegen, damit ich es nicht vergesse.

Ich als Dozent will Notizen löschen, damit ich sie nicht sehe.

Ich als Dozent will Notizen bearbeiten, damit sie nicht falsch bleiben.

Administrator:

Ich als Administrator will Fragen löschen, damit das Forum respektvoll bleibt.

Ich als Administrator will Fragen modifizieren, damit sie nicht falsch bleiben.

Ich als Administrator will Konten von anderen löschen, damit das Forum respektvoll bleibt.

Ich als Administrator will Fragen sehen, damit ich alles kontrollieren kann.

Ich als Administrator will Beiträge sehen, damit ich alles kontrollieren kann.

Ich als Administrator will Neuigkeiten stellen, damit alle das sehen können.

Rahmenbedingungen:

Die App soll eine Datenbank benutzen.

Die App soll auf mehr als 90% Geräte gut arbeiten können.

Tabellarische Darstellung + ggf. zukünftige User-Stories:

<i>Anmeldung</i>	
ID	1
Priorität	Core
Beschreibung	Ein Benutzer kann sich über seine Uni-Zugangsdaten registrieren
Ergänzungen	+ Anbindung an das Hochschul-Portal

Wireframe / UI-Skizze	Es wird ein E-Mail- und Passwortfeld angezeigt, in die ein Benutzer seine Daten eintragen kann. Über einen „Anmelden“ Button wird der Zugang geprüft
Entwickleraufgaben	<ul style="list-style-type: none"> - FireBase DB - Formularfelder erstellen - Datenabgleich - Daten in DB schreiben - Eingabepprüfung (Syntax) - Versenden einer autom. Mail

<i>Beiträge erstellen</i>	
ID	2
Priorität	Core
Beschreibung	Ein Benutzer kann Beiträge in Foren erstellen
Ergänzungen	<ul style="list-style-type: none"> + Formatierungsmöglichkeiten anbieten + Einbettung von Links + Einbettung von Videos (als Link, ggf. passende App öffnen) + Einbettung von Bildern (+ max. Größe def.) + Einbettung von Dateien (+ max. Größe def.)
Wireframe / UI-Skizze	Ein Beitrag wird über das + Symbol hinzugefügt; Jede Formatierungsmöglichkeit hat ihre eigenen typischen Icons; Der eigentliche Text wird in einem darunterliegenden Textfeld hinterlegt. Über ein Iconn wird der Beitrag veröffentlicht
Entwickleraufgaben	<ul style="list-style-type: none"> - Formatierungsmöglichkeiten als HTML-Tags abbilden - Click-Listener auf Icons - Dateupload erlauben (→ Rechte) - Limits für max. Größen in Konstanten definieren - Erlaubte Formate in Konstanten definieren - Formularfelder erstellen - Buttons mit Click-Listnern belegen - Video-Einbettung konfigurieren (Darstellung?)

	<ul style="list-style-type: none"> - Beitragsveröffentlichung mit bisheriger Beitragshistorie synchronisieren - Beitrag in DB sichern
--	---

Beiträge bewerten

ID	3
Priorität	Extension
Beschreibung	Ein Benutzer kann durch ein Klick auf Daumen nach oben Beiträge und Themen bewerten. Relevante Themen rutschen weiter nach oben
Ergänzungen	+ Icons erstellen/raussuchen
Wireframe / UI-Skizze	Daumen nach oben Icon pro Beitrag; Animation beim jeweiligen auf- und hinabrutschen von Beiträgen
Entwickleraufgaben	<ul style="list-style-type: none"> - ClickListener auf Icon - Animation definieren - Abgleich der Daumen hoch Zahl im jeweiligen Thema ⇔ ggf. Verschiebung eines Beitrags → Beiträge neu anordnen - Anzahl Daumen hoch in DB sichern

Beiträge melden

ID	4
Priorität	Core
Beschreibung	Ein Benutzer kann anstößige Inhalte melden
Ergänzungen	+ Admins erhalten Mitteilung per Nachricht und/oder E-Mail
Wireframe / UI-Skizze	Es wird pro Beitrag ein kleines anklickbares Ausrufezeichen angezeigt; Beim Klick darauf wird ein Dialog geöffnet, der wiederum bestätigt werden muss
Entwickleraufgaben	<ul style="list-style-type: none"> - ClickListener auf Icon → Dialog öffnen - Bei erneuter Bestätigung → autom. Mail/Mitteilung an Admin senden

Stundenplan abrufen

ID	5
Priorität	Extension
Beschreibung	Ein Benutzer kann seinen Stundenplan über einen eigenen Reiter abrufen
Ergänzungen	+ Anbindung an das LSF der Hochschule
Wireframe / UI-Skizze	Reiter wird mit Icon und Text in der Sidebar platziert. Alternativ ggf. indirekter Zugang über den LSF-Reiter
Entwickleraufgaben	<ul style="list-style-type: none"> - Icon + Text in Sidebar legen - ClickListener auf Icon → LSF Startseite über internen Browser laden

Mensaplan abrufen

ID	6
Priorität	Extension
Beschreibung	Ein Benutzer kann den Mensaplan der Hochschule über eigenen Reiter abrufen
Ergänzungen	
Wireframe / UI-Skizze	Reiter wird mit Icon und Text in der Sidebar platziert
Entwickleraufgaben	<ul style="list-style-type: none"> - Icon + Text in Sidebar legen - ClickListener auf Icon → Mensaplan über internen Browser laden <i>oder</i> auslesen und in Tabelle darstellen

Beiträge löschen

ID	7
Priorität	Core
Beschreibung	1) Ein Benutzer kann seine eigenen Beiträge löschen 2) Ein Admin kann jeden Beitrag löschen
Ergänzungen	+ ggf. autom. Mitteilung an Benutzer, falls Admin Beitrag löscht
Wireframe / UI-Skizze	1) Beitrag löschen über Button „Edit“ → Link „Beitr. löschen“ 2) Admins klicken auf einen Link „Beitrag entf.“

Entwickleraufgaben	<ul style="list-style-type: none"> - ClickListener auf „Edit“ → Edit-Ansicht - ClickListener auf „Beitr. löschen/entf.“ → Beitr. aus DB entf. - Beitragshistorie aktualisieren - Beitragsansicht neu laden
---------------------------	--

Unterforen auswählen

ID	8
Priorität	Core
Beschreibung	Ein Benutzer kann sich in beliebig vielen Unterforen über Buttons anmelden
Ergänzungen	+ Icon nach Klick verändern
Wireframe / UI-Skizze	Realisierung über eine Glocke, die als Subscribe-Button agiert; Glocke wandelt sich nach Klick in ein anderes Icon
Entwickleraufgaben	<ul style="list-style-type: none"> - ClickListener auf Icon → gesicherte Foren in Liste ablegen - Zugriff über eigenen Reiter ⇔ Liste auslesen - Shortcuts definieren - User-Konfiguration in DB schreiben

Neue Unterforen anlegen

ID	9
Priorität	Core
Beschreibung	Ein Admin kann neue Unterforen über einen Button anlegen und konfigurieren
Ergänzungen	+ Konfiguration auch nachträglich möglich
Wireframe / UI-Skizze	Ein Button „Neues Forum erstellen“ wird dem Admin in der Forenübersicht angezeigt; Die Einstellungen dazu sind ähnlich wie die der User-Story „Beiträge erstellen“
Entwickleraufgaben	<ul style="list-style-type: none"> - ClickListener auf Button → leicht angepasste Edit-Sicht (s.o.) - Unterforum in bereits ex. Hierarchie eingliedern - Auswahl bestimmter Optionen (?) - Neues Forum mit DB bekannt machen

Teilnehmer benachrichtigen

ID	10
Priorität	Extension
Beschreibung	Tutoren und Dozenten können auswählen, ob sie alle Teilnehmer eines Forums (außer sich selbst) beim Abschicken ihres Beitrags benachrichtigen wollen
Ergänzungen	
Wireframe / UI-Skizze	Beim verfassen des Beitrags wird unter dem Eingabetextfeld eine Checkbox „alle Teilnehmer benachrichtigen“ angezeigt
Entwickleraufgaben	<ul style="list-style-type: none"> - ClickListener auf Checkbox → Variable setzen - Event bei Beitragsveröffentlichung auslösen - Autom. E-Mail/Mitteilung an alle Beteiligten senden

Notizen erstellen & abrufen

ID	11
Priorität	Extension
Beschreibung	Ein Benutzer kann Notizen über einen eigenen Reiter erstellen, speichern und abrufen
Ergänzungen	
Wireframe / UI-Skizze	„Notes“ Reiter wird mit Icon und Text in der Sidebar platziert. Einzelne Notizen werden in untereinanderliegende UI-Cluster-Elemente abgelegt und können per Wischen bedient werden
Entwickleraufgaben	<ul style="list-style-type: none"> - ClickListener pro Icon - Formatierungsmöglichkeiten als HTML-Tags abbilden - Notiz(en) in DB schreiben/lesen - Autom. Speicherung alle x-Sekunden - Wischfunktionalität erstellen - Geeignetes Widget/View wählen

Private Textnachrichten

ID	12
Priorität	Extension

Beschreibung	Ein Benutzer kann andere Benutzer privat über einen eigenen Chat kontaktieren
Ergänzungen	+ Blockierfunktion
Wireframe / UI-Skizze	Durch Klick auf ein Benutzerprofilbild gelangt man auf ein anderes Profil und wählt das Chat-Icon
Entwickleraufgaben	<ul style="list-style-type: none"> - ClickListener auf Profilbild → Shortcut auf Profilseite - ClickListener auf Icon → Fragment öffnen - Textfeld einrichten - ClickListener auf Absende-Icon → Text aus Textfeld im Chat darstellen - Chatverlauf in DB sichern

Profileinstellungen

ID	13
Priorität	Extension
Beschreibung	Ein Benutzer kann über einen eigenen Reiter einfache Profileinstellungen vornehmen
Ergänzungen	<ul style="list-style-type: none"> + Bild hochladen/ändern + Beschreibung hinzufügen + Studiengang ändern + Alter ändern/angeben + Semester ändern
Wireframe / UI-Skizze	„Profil“ Reiter wird mit Icon und Text in der Sidebar platziert; Durch ein Klick auf das Profilbild unter „Settings“, lässt sich ein anderes auswählen. Andere Einstellungen können weiter unten direkt in Textfeldern editiert oder über Checkboxes angekreuzt werden; Über den „Save“ Button werden die Einstellungen gesichert
Entwickleraufgaben	<ul style="list-style-type: none"> - Click-Listener auf Icons - Dateiapload erlauben (→ Rechte) - Limits für max. Größen in Konstanten definieren - Erlaubte Formate in Konstanten definieren - Formularfelder/Checkboxes erstellen - Buttons mit ClickListnern belegen

- Konfigurationen in DB sichern

App-Einstellungen

ID	14
Priorität	Extension
Beschreibung	Ein Benutzer kann über einen eigenen Reiter einfache Änderungen an der App vornehmen
Ergänzungen	<ul style="list-style-type: none"> + Dark-Mode + Schriftgröße + Autostart festlegen + Speicherung der LogIn-Daten festlegen + Art der Benachrichtigung festlegen
Wireframe / UI-Skizze	„Settings“ Reiter wird mit Icon und Text in der Sidebar platziert; Einstellungen können entweder über Buttons oder Checkboxes angekreuzt werden
Entwickleraufgaben	<ul style="list-style-type: none"> - s.o. ([-) Dateiupload [-] erlaubte Formate [-] Limits def.)

Beitragsentwurf erstellen

ID	15
Priorität	Extension
Beschreibung	Ein Benutzer kann beim Verfassen eines Beitrags ihn über einen Button als Entwurf für später speichern
Ergänzungen	+ Entwurf in DB zwischenspeichern
Wireframe / UI-Skizze	Zwischenspeichern über Button „Entwurf“; Entwurf wird ausgegraut im jeweiligen Forum angezeigt und kann über den Button „Entwurf veröffentlichen“ veröffentlicht werden
Entwickleraufgaben	<ul style="list-style-type: none"> - ClickListener auf Button → Beitrag ausgegraut einblenden - Beitrag temporär in DB sichern - Alte Entwürfe nach x-Tagen aus DB entf. - Ggf. Reiter + Icon für Entwürfe + ClickListener

Systemnachricht

ID	16
Priorität	Core
Beschreibung	Ein Admin kann über einen eigenen Reiter eine Systemnachricht verfassen (z.B. Hinweis zu Wartungsarbeiten), welche allen Benutzern zentral angezeigt wird
Ergänzungen	
Wireframe / UI-Skizze	Reiter wird mit Icon und Text in der Sidebar platziert; Systemnachricht wird anschließend über ein Textfeld eingegeben und über einen „Veröffentlichen“ Button abgeschickt
Entwickleraufgaben	<ul style="list-style-type: none"> - ClickListener auf Reiter + Icon → neues View - Formularfelder/Checkboxen erstellen - Autom. E-Mail & interne Benachrichtigung an alle - Ggf. globales View mit Nachricht anzeigen ⇔ in DB schreiben

Beiträge markieren

ID	17
Priorität	Extension
Beschreibung	Ein Benutzer kann Beiträge, die er später nochmal anschauen möchte per Klick auf einen Button neben jedem Beitrag markieren und erstellt damit einen Shortcut im Reiter
Ergänzungen	+ Entwurf in DB zwischenspeichern
Wireframe / UI-Skizze	Beiträge werden über ein Herz-Icon pro Beitrag angezeigt. Bei einem Klick auf das Herz, wird eine Animation ausgelöst. Unter dem Reiter „Gemerkt“ in der Sidebar mit Icon und werden anschließend alle Beiträge untereinander angezeigt
Entwickleraufgaben	<ul style="list-style-type: none"> - ClickListener auf Icon → Animation → in Liste schreiben - Reiter „Gemerkt“ mit ClickListener belegen → neues View - Liste auslesen - Shortcut auf Beitrag für jedes Listenelement

Lösungsbeitrag markieren

ID	18
Priorität	Extension
Beschreibung	Ein Benutzer kann in seinem eigenen Thema eine für ihn funktionierende Lösung (Beitrag) über ein Icon markieren. Das Thema bekommt den Status „Gelöst“ und wird geschlossen
Ergänzungen	
Wireframe / UI-Skizze	Pro Beitrag ex. ein Icon, welches sich anklicken lässt. Ist ein Thema gelöst, wird in der Themenübersicht ein Schloss-Icon angezeigt
Entwickleraufgaben	<ul style="list-style-type: none"> - Click-Listener auf Icon - Forum schließen → Variable (= Status) in DB ändern → Buttons ausblenden + Schloss-Icon einblenden

Autom. nachladen von Beiträgen

ID	19
Priorität	Core
Beschreibung	Ein Benutzer bekommt neue Beiträge in Echtzeit angezeigt
Ergänzungen	
Wireframe / UI-Skizze	Neuer Beitrag wird an oberster Stelle mit einer Animation + Geräusch angezeigt
Entwickleraufgaben	<ul style="list-style-type: none"> - Autom. Aktualisierung (?) - Event löst Sound + Animation aus

Vorschau für Themen und Beiträge

ID	20
Priorität	Extension
Beschreibung	Ein Benutzer kann sich vor der Veröffentlichung eines Themas und Beitrags über einen Button eine Vorschau generieren lassen
Ergänzungen	+ Eingliederung in das Forum, ohne Sichtbarkeit für andere
Wireframe / UI-Skizze	Anzeige über den Button „Vorschau“ generieren. Beitrag wird anschließend etwas ausgegraut in der Beitragshierarchie eingegliedert

**Entwickleraufga
ben**

- Vgl. User-Story „Beitragsentwurf erstellen“ ([-]
zwischenspeichern in DB)