



universität
wien

BACHELORARBEIT

ENRICHED DIDITAL DIARY

Verfasser

Mehremic Adem

angestrebter akademischer Grad

Bachelor of Science (BSc)

Wien, 2022/2023

Studienkennzahl lt.
Studienblatt:

A 01650669<

Fachrichtung:

Informatik - Allgemein

Betreuerin / Betreuer:

Dr. techn. Belal Abu Naim

1. Motivation	4
2. Verwandte Arbeit.....	5
2.1. Analyzing	5
2.2. Apps	5
2.2.1. Five Minute Journal	5
2.2.2. Day One	6
2.2.3. Penzu.....	6
2.2.4. Journey.....	6
3. Design und Major Idea	8
3.1. Benutzerauthentifizierung: Registrierungs- und Anmeldefunktion	9
3.2. Database	10
3.3. Entry Erstellung.....	11
3.4. Entry Darstellung	14
3.5. Standort	16
3.6. Analyse und Kategorisierung mit ML Kit.....	16
3.7. Listing	17
4. Implementation	19
5. Future Work	30
6. Schlussfolgerung	32
7. Biography	33

Abstract

E-Diary ist eine digitale Tagebuch-App für Android-Benutzer, die das alltägliche normale Journaling verändert. Es erleichtert dem Benutzer, seine Ideen jeden Tag auf multidimensionale Weise festzuhalten, eine Geschichte zu erzählen und seinem Dokument mehr als nur Text hinzuzufügen. Heutzutage haben die Menschen ein hektisches Leben und viele Hindernisse, um sich ganz dem Tagebuch zu widmen, E-Diary macht es einfacher und ist eine perfekte Alternative zu Zeitbeschränkungen. Mit E-Diary können die Benutzer Einträge schreiben, denen auch Bilder hinzugefügt werden, und bei ihrer Erstellung auch mit Geotags versehen werden und mit maschine learning analysiert und kategorisiert. Die App bietet auch eine bemerkenswerte Funktion, das Sicherheitsschloss, das Passcode- und Fingerabdrucksicherheit umfasst, sodass niemand die Daten ohne die Erlaubnis des Benutzers sehen kann.

1. Motivation

Diaring ist ein mächtiges Werkzeug, das sich nachweislich positiv auf das geistige und emotionale Wohlbefinden auswirkt [1] und steigert die Lebenszufriedenheit und das Glück [2]. Trotz der Vorteile führt jedoch weniger als ein Fünftel der Menschen tatsächlich Tagebuch[3]. Ein Tagebuch zu führen kann eine herausfordernde Aufgabe sein, besonders für diejenigen, die neu in der Praxis sind. Es erfordert Disziplin, Zeit und Mühe, alltägliche Erfahrungen aufzuschreiben und darüber nachzudenken. Hier kommt die Enriched Digital Diary-App ins Spiel.

Die Idee hinter diesem Projekt war es ein digitales Tagebuch zu erstellen, wo die Benutzer ihre Gedanken, Gefühle, Erlebnisse und anderen Informationen aufnehmen und Speichern können. Das Tagebuch an sich ist uns alles bekannt als ein Buch oder ein Heft wo man seinen Gedanken freien lauf lassen kann. Da wir in einer Zeit leben wo die Digitalisierung alles rund um uns übernimmt, so war es für diese Arbeit wichtig ein Tagebuch zu digitalisiert und Features zu integrieren welche nur mit unserem Smartphone oder einer mobilen Applikation möglich sind. Digitale Tagebücher kann man in der heutigen Zeit schon einige finden, jedoch war es wichtig das eine App entwickelt wird welche sich von den anderen unterscheidet.

Ein Punkt welcher uns sehr wichtig war, war die Sicherheit unserer Daten welche mit Fingerabdrucksensor oder einer Passwort geregelt ist, so das nur der User bzw. Besitzer der Tagebuchs zugriff hat. Die Applikation ermöglicht es den User Texte hinzufügen und auch Bilder zu den jeweiligen Texten. Jedoch ist der spannendste Teil dieser Arbeit das integrierte Machine Learning welches die aufgenommenen Bilder analysiert anhand einer bestehenden Datenbank an Informationen. So werden den aufgenommenen Bilder "Tags" hinzugefügt, welche die Bilder genau beschreiben. Die ganze Applikation ist darauf ausgerichtet das der User ein Tagebuch mit Bildern und Machinelearning in einem hat und so alles in einem Ort hat. Die App ist einfach zugänglich, sicher und auch leicht zu benutzen. Mit der Motivation wurde auch diese Arbeit gemacht.

2. Verwandte Arbeit

2.1. Analyzing

Bei der Recherche verwandter Arbeiten für das angereicherte digitale Tagebuch war es wichtig, sich bestehende Tagebuch- und Journalanwendungen anzusehen, die ähnliche Funktionen und Merkmale bieten. Dazu gehören Funktionen wie die Möglichkeit, tägliche Einträge vorzunehmen, Fotos und Multimedia hinzuzufügen, Inhalte zu kategorisieren und zu analysieren und Tools zur Reflexion und Selbstverbesserung anzubieten.

Ein wichtiger Aspekt dieser Recherche besteht darin, sowohl die Ähnlichkeiten als auch die Unterschiede zwischen der verwandten Arbeit und der App zu identifizieren. Ähnliche Merkmale könnten eine benutzerfreundliche Oberfläche, die Möglichkeit zum Hinzufügen und Bearbeiten von Einträgen und ein Fokus auf persönliches Wachstum und Selbstreflexion umfassen. Zu den Unterschieden können einzigartige Funktionen wie die Verwendung von maschinellem Lernen oder Bilderkennung oder ein robusterer Satz von Anpassungs- und Analysetools gehören.

Da es in der Arbeit um eine Smartphone App gehen wir auch unsere Suche nur Related Works abgegrenzt auf weitere Smartphone Apps im Google Store und Apple App Store.

Die naheliegendsten Apps die wir in unserer Analyse gefunden haben wären "Five Minute Journal", "Day One", "Penzu" und "Journey". Diese Apps ermöglichen es dem User Text-Basierte Einträge zu erstellen, und enthalten auch Funktionalitäten wie Fotos und Audioaufnahmen zu hinzufügen. Es gibt auch viele andere Applikationen die mit wenig Arbeit in unserer App zu einem Related work führen konnte. Das wären z.B:

2.2. Apps

2.2.1. Five Minute Journal

Das Five Minute Journal ist eine beliebte Anwendung zur Selbstverbesserung und Achtsamkeit, die Benutzern hilft, ihren Tag mit Dankbarkeit zu beginnen und ihn mit Reflexion zu beenden. Es basiert auf dem Konzept der positiven Psychologie und ermutigt die Benutzer, sich auf den gegenwärtigen Moment zu konzentrieren und Dankbarkeit zu üben. Die App bietet Benutzern Aufforderungen, drei Dinge aufzuschreiben, für die sie dankbar sind, drei Dinge, die den Tag großartig machen würden, und eine tägliche Bestätigung. Am Ende des Tages werden die Nutzer

gebeten, über drei Dinge nachzudenken, die gut gelaufen sind und wie sie den Tag noch besser hätten machen können.

2.2.2. Day One

Day One ist eine beliebte Tagebuchanwendung, mit der Benutzer ihre täglichen Erfahrungen, Gedanken und Erinnerungen aufzeichnen können. Es ermöglicht Benutzern, ihre Einträge mit einer Reihe von Funktionen wie Text, Fotos und Tags einfach zu erstellen und zu speichern. Die App bietet auch eine Reihe von Anpassungsoptionen wie Schriftartauswahl, Hintergrundthemen und Passwortschutz. Eines der Hauptmerkmale von Day One sind die Einträge mit Zeitstempel, die es den Benutzern ermöglichen, den Kontext und den Fortschritt ihrer Aufzeichnungen im Laufe der Zeit zu sehen. Darüber hinaus lässt sich Day One in verschiedene andere Apps wie Apple Health und Fitbit integrieren, sodass Benutzer ihre Fitness- und Gesundheitsdaten neben ihren Tagebucheinträgen einfach verfolgen können.

2.2.3. Penzu

Penzu ist eine digitale Journaling-Plattform, die Benutzern einen sicheren und privaten Raum bietet, um ihre Gedanken, Gefühle und Erinnerungen festzuhalten. Es bietet Benutzern eine Schnittstelle zum Schreiben und Organisieren ihrer Einträge mit Funktionen wie anpassbaren Hintergründen und Schriftarten, Passwortschutz und der Möglichkeit, Fotos und Audio hinzuzufügen. Penzu bietet auch eine mobile App, mit der Benutzer von überall aus auf ihr Tagebuch zugreifen können. Ein einzigartiger Aspekt von Penzu ist die „Time Capsule“-Funktion, die es Benutzern ermöglicht, ihre Tagebucheinträge zu sperren und erst nach Ablauf einer bestimmten Zeit darauf zuzugreifen.

2.2.4. Journey

Journey ist eine beliebte digitale Tagebuchanwendung, die auf mehreren Plattformen verfügbar ist, darunter Web, iOS und Android. Es bietet eine benutzerfreundliche Oberfläche, die es Benutzern ermöglicht, ihre täglichen Gedanken, Erinnerungen und Erfahrungen auf organisierte Weise einfach zu erfassen und aufzuzeichnen. Die App bietet eine Reihe von Funktionen wie anpassbare Vorlagen, automatische Fotosicherung, Passwortschutz und Integration mit Cloud-Diensten. Journey bietet Benutzern auch die Möglichkeit, Multimedia-Elemente wie Fotos, Videos und Audioaufnahmen zu ihren Einträgen hinzuzufügen, was es zu einem großartigen Werkzeug für den kreativen Ausdruck macht. Darüber hinaus bietet Journey eine Funktion namens „Journey Insights“, die

aussagekräftigen Einblicke und Statistiken basierend auf den Tagebucheinträgen des Benutzers generiert. Dies kann Informationen über ihre täglichen Stimmungen, Muster in ihrem Schreiben und sogar das Wetter während der Tage beinhalten, an denen sie geschrieben haben.

Diese verwandten Werke haben einige gemeinsame Merkmale mit der erweiterten digitalen Tagebuchanwendung, wie z. B. die Möglichkeit, ein persönliches Tagebuch zu führen, Fotos hinzuzufügen und Standortinformationen zu verfolgen. Die angereicherte digitale Tagebuch-App unterscheidet sich jedoch durch den Einsatz von maschinellem Lernen, um Bilder zu kategorisieren und zu analysieren und ein umfassenderes und personalisierteres Tagebucherlebnis zu bieten.

3. Design und Major Idea

Die E-Diary Applikation wurde entwickelt, um Benutzern eine umfassende und interaktive Möglichkeit zu bieten, ihre Lebenserfahrungen aufzuzeichnen und zu organisieren. Diese Funktionen sollen Benutzern eine bequeme und intuitive Möglichkeit bieten, ihre Erfahrungen zu reflektieren und zu organisieren.

In diesem Abschnitt möchten wir einen umfassenden Überblick über die zugrunde liegenden Konzepte, Ideen und das Design hinter den verschiedenen Funktionalitäten der Anwendung geben. Um dies zu erreichen, verwenden wir eine Kombination aus Klassendiagrammen, Sequenzdiagrammen und Entwurfsskizzen. Dies wird dazu beitragen, die verschiedenen Komponenten und Interaktionen zwischen ihnen zu veranschaulichen und ein klares Verständnis dafür zu vermitteln, wie die verschiedenen Funktionen der Anwendung strukturiert und implementiert wurden. Darüber hinaus werden Designbeispiele vorgestellt, um die Implementierung und Verwendung der verschiedenen Funktionalitäten weiter zu demonstrieren. Insgesamt besteht das Ziel darin, eine klare und detaillierte Erläuterung des Anwendungsdesigns bereitzustellen, damit die Leser die Funktionsweise der Anwendung besser verstehen und wie sie entwickelt wurde, um bestimmte Anforderungen zu erfüllen und bestimmte Probleme zu lösen.

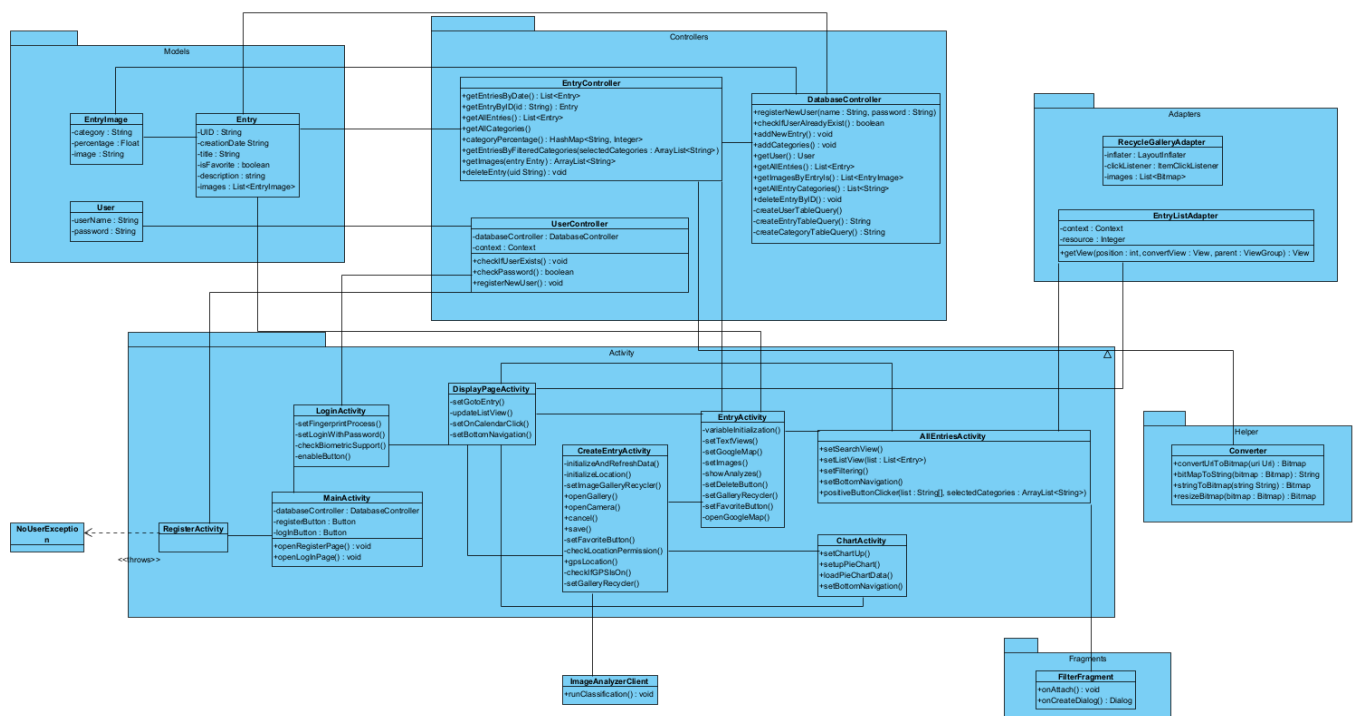


Figure 1 Class Diagram of the application

3.1. Benutzerauthentifizierung: Registrierungs- und Anmeldefunktion

Die Registrierungs- und Login-Funktionalität sind Schlüsselkomponenten der Anwendung, da sie die Sicherheit und Vertraulichkeit der Informationen und Tagebucheinträge der Benutzer gewährleisten, da der Nutzer jeden Tag verschiedenen Datenschutzrisiken, bei der Verwendung mobiler Apps ausgesetzt sind [4]. Der Registrierungsprozess umfasst die Erstellung eines neuen Kontos durch Angabe persönlicher Daten wie Benutzername und Passwort. Der Anmeldevorgang ermöglicht Benutzern dann den Zugriff auf ihr Konto durch Eingabe ihres Benutzernamens und Passworts oder, für zusätzliche Sicherheit, durch Authentifizierung per Fingerabdruck [5]. Diese Funktionen geben Benutzern die Gewissheit, dass ihre persönlichen Informationen und Tagebucheinträge vertraulich behandelt werden und nur für sie selbst zugänglich sind. Durch die Kombination dieser beiden Funktionalitäten bietet die Anwendung eine sichere Plattform für Benutzer, um ihre täglichen Erfahrungen zu reflektieren und zu dokumentieren.

Register

Die Zukunftsvision für diese Anwendung besteht darin, sie in eine serverbasierte Plattform umzuwandeln, bei der jeder Aspekt der Anwendung mit einem Benutzerkonto verknüpft ist. Um diesen Übergang zu ermöglichen, werden die notwendigen Grundlagen geschaffen. Daher werden Benutzer, die die Anwendung installieren, aufgefordert, ein Benutzerkonto zu erstellen und zu verwenden. Es ist wichtig zu beachten, dass jedes Gerät nur ein registriertes Benutzerkonto haben kann. Falls ein Benutzer versucht, mehrere Konten zu registrieren, wird eine Popup-Meldung angezeigt, die besagt: „Ein Benutzerkonto existiert bereits, bitte melden Sie sich an“. Dieses System wird eingerichtet, um eine optimierte und personalisierte Benutzererfahrung zu gewährleisten.

Login

Nachdem der Benutzer sein Konto erfolgreich erstellt hat, kann er sich über eine sichere Anmeldeseite beim digital angereicherten Tagebuch anmelden. Die Anmeldeseite enthält Optionen für die Kennwort- und Fingerabdruckauthentifizierung. Die Passwortoption ermöglicht es Benutzern, ihr Passwort einzugeben und auf ihr Tagebuch zuzugreifen, während die Fingerabdruckoption biometrische Erkennung verwendet, um den Zugriff zu gewähren. Die Fingerabdruck-Option bietet eine schnelle und bequeme Möglichkeit, auf das Tagebuch zuzugreifen, ohne sich ein Passwort merken zu

müssen. Die Passwort- und Fingerabdruckoptionen wurden entwickelt, um Benutzern eine sichere und bequeme Möglichkeit zu bieten, auf ihr Tagebuch zuzugreifen und ihre darin gespeicherten persönlichen Informationen und Erinnerungen zu schützen.

The figure displays three mobile application screens for the 'E-Diary' app. Each screen has a status bar at the top with the time and signal indicators, and a brown header bar with the app name 'E-Diary'.

- Screen 1 (Left): 'Set up Account'**
 - Title: *Set up Account*
 - Form fields: 'name', 'Enter password', and 'Repeat password'.
 - Button: A brown 'CREATE' button at the bottom.
- Screen 2 (Middle): 'Log-In'**
 - Title: *Log-In*
 - Form field: 'Enter password'.
 - Buttons: Two brown buttons labeled 'LOGIN WITH PASSWORD' and 'LOGIN WITH FINGERPRINT'.
- Screen 3 (Right): 'Log-In'**
 - Title: *Log-In*
 - Form field: 'Enter password'.
 - Buttons: Two grey buttons labeled 'LOGIN WITH PASSWORD' and 'LOGIN WITH FINGERPRINT'.
 - Footer: A white button with a green icon and text 'Please Register a User.'.

Each screen features a black navigation bar at the bottom with three icons: a back arrow, a home circle, and a recent apps square.

Figure 2 Design of the Register and Login page

3.2. Database

Die betrachtete Anwendung für digital angereicherte Tagebücher speichert und verwaltet Daten lokal auf dem Telefon des Benutzers, ohne auf einen zentralen Server angewiesen zu sein. Angesichts der Natur dieser Anwendung als langfristiges Werkzeug, von dem erwartet wird, dass es eine beträchtliche Datenmenge in Form von Tagebucheinträgen, Bildern, Text und analysierten Informationen sammelt, wurde es als wichtig erachtet, ein robustes und effizientes Datenstruktursystem auszuwählen diese Aufgaben zu bewältigen.

Angesichts dieser Anforderungen entschieden sich die Entwickler der Anwendung für SQLite als Datenbankmanagementsystem. [6] SQLite ist eine weit verbreitete Open-Source-Softwarebibliothek, die ein leichtes, dateibasiertes Datenbankverwaltungssystem bereitstellt. Es hat einen geringen Platzbedarf und ist daher ideal für den Einsatz auf mobilen Geräten und anderen Systemen mit begrenzten Speicher- und Arbeitsspeicherressourcen [7]. Darüber hinaus ist SQLite in der Lage, den gleichzeitigen Zugriff von mehreren Prozessen zu handhaben, wodurch es sich gut für Anwendungen wie ein digitales Tagebuch eignet, auf das der Benutzer jederzeit zugreifen kann [8].

Die Verwendung von SQLite in diesem Zusammenhang entspricht den aktuellen Trends in der Softwareentwicklung zur Verwendung leichtgewichtiger, dateibasierter Datenbanken für die lokale Datenspeicherung. Dies ist besonders relevant im Bereich der Verwaltung personenbezogener Daten, wo der Datenschutz und die Sicherheit der Benutzer von entscheidender Bedeutung sind.

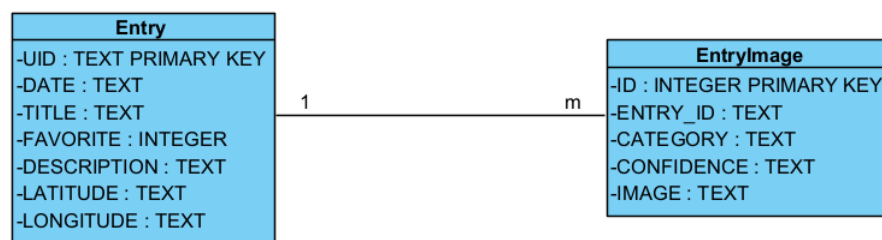


Figure 3 Daten Struktur und Beziehung von Entry und EntryImage

3.3. Entry Erstellung

Die Erstellungsseite für Tagebucheinträge in der angereicherten Tagebuchanwendung bietet eine benutzerfreundliche und intuitive Schnittstelle zum Aufzeichnen und Organisieren ihrer Erfahrungen und Erinnerungen.

Der Benutzer beginnt mit der Eingabe eines Titels für den Eintrag, der als kurze Zusammenfassung der aufgezeichneten Erfahrung oder Erinnerung dient. Die Option, den Eintrag als Favorit zu markieren, ermöglicht es dem Benutzer, seine schönsten Erlebnisse und Erinnerungen leicht zu identifizieren und schnell darauf zuzugreifen.

Der Benutzer hat dann die Möglichkeit, ein Bild aus seiner Galerie auszuwählen oder ein neues Bild mit der Kamera aufzunehmen, das dann einer horizontalen Galerieansicht hinzugefügt wird. Die horizontale Galerieansicht bietet eine bequeme Möglichkeit, die mit dem Eintrag verknüpften Bilder zu überprüfen und zu verwalten, und wenn Sie auf ein Bild in der Galerieansicht klicken, wird das Bild in der Hauptbildansicht angezeigt.

Die Anwendung bietet auch die Möglichkeit, den aktuellen Standort zu speichern, was ermöglicht wird, indem man die Erlaubnis des Benutzers einholt und sicherstellt, dass GPS eingeschaltet ist. Diese Funktion ermöglicht es dem Benutzer, seine Erfahrungen und Erinnerungen mit bestimmten Orten zu verknüpfen und so eine reichhaltige und detaillierte Aufzeichnung seines Lebens bereitzustellen.

Schließlich hat der Benutzer die Möglichkeit, eine Textbeschreibung hinzuzufügen, die weiteren Kontext und Details über die aufgezeichnete Erfahrung oder Erinnerung liefert. Diese Beschreibung kann verwendet werden, um über die Bedeutung des Eintrags nachzudenken oder um ein differenzierteres Verständnis der Erfahrung oder Erinnerung zu vermitteln.

Insgesamt bietet die Erstellungsseite eine umfassende und intuitive Möglichkeit für eine Person, ihre Erfahrungen und Erinnerungen zu erfassen, zu organisieren und darüber nachzudenken, was sie zu einer Schlüsselkomponente der angereicherten Tagebuchanwendung macht.

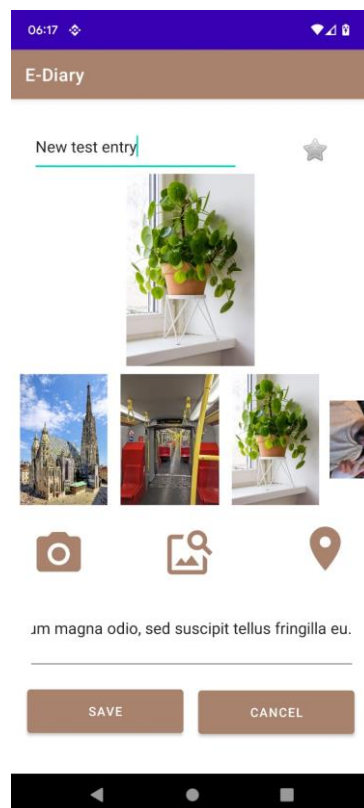


Figure 4 Entry erstellungs Seiten design

3.4. Entry Darstellung

Die Eintragsanzeigeseite in der E-Diary App soll einen umfassenden Überblick über jeden Tagebucheintrag bieten. Die Seite enthält den Eintragstitel, die Hauptbildansicht und eine horizontale Galerieansicht, die alle dem Eintrag zugeordneten Bilder anzeigt. Der Benutzer kann auf ein beliebiges Bild in der Galerie klicken, um es in der Hauptbildansicht anzuzeigen. Die Seite enthält auch die Eintragsbeschreibung, die zusätzlichen Kontext und Details zum Eintrag bereitstellt. Der Eintrag kann als Favorit markiert werden, wodurch der Benutzer seine wichtigsten Einträge schnell identifizieren kann. Zusätzlich zeigt die Anzeigeseite des Eintrags das Datum, an dem der Eintrag erstellt wurde, sowie die Beschreibung.

Die Seite verfügt außerdem über drei Schaltflächen: Löschen, mit dem der Benutzer den Eintrag löschen kann; Kategorisierung anzeigen, die die Kategorisierungsergebnisse für jedes Bild einzeln anzeigt; und Standort in Google Maps öffnen, wodurch der Standort des Eintrags in Google Maps geöffnet wird, wenn der Benutzer der App die Erlaubnis erteilt hat, auf seinen Standort zuzugreifen, und sein GPS eingeschaltet hat. Die Eintragsanzeigeseite ist so konzipiert, dass sie eine klare und intuitive Möglichkeit bietet, Tagebucheinträge anzuzeigen und zu verwalten, wodurch der Zugriff auf und die Organisation der Informationen erleichtert wird.

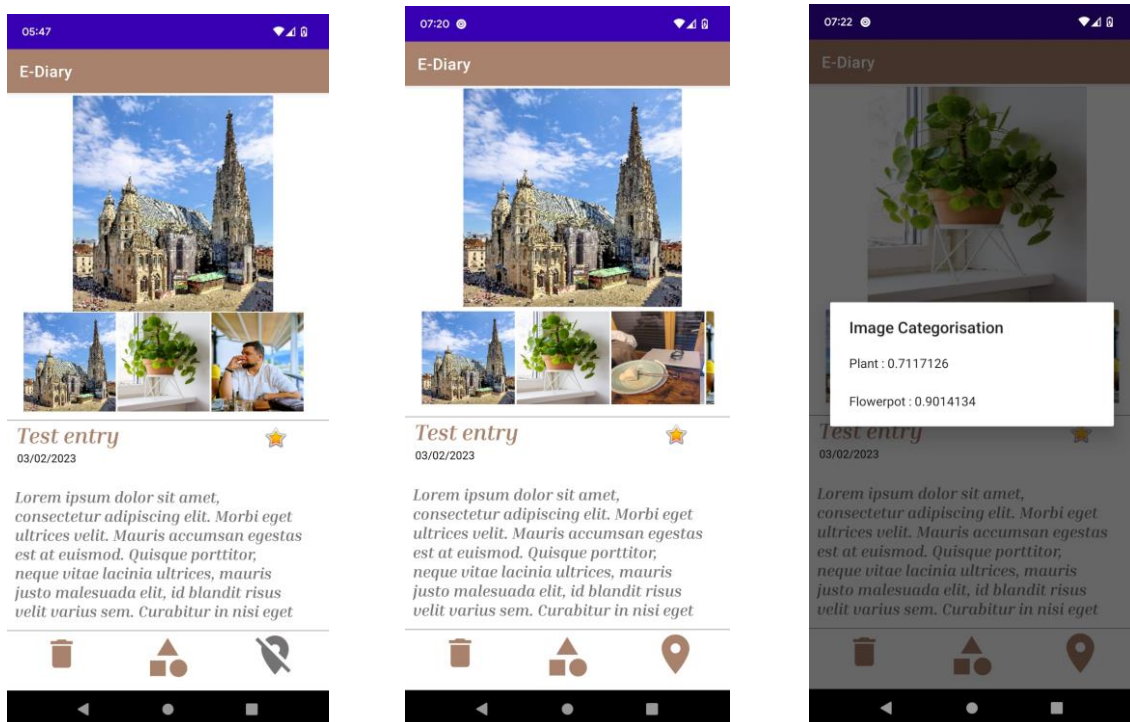


Figure 5 Entry Darstellungs Seite Design

3.5. Standort

Die Standortfunktion in dieser App ermöglicht es Benutzern, den Ort zu sehen, an dem ihre Eingaben vorgenommen wurden, indem sie ihn in Google Maps öffnen. Diese Funktion bietet Benutzern eine visuelle Darstellung dessen, wo sie sich befanden, als sie einen bestimmten Eintrag vorgenommen haben, und fügt ihrem Tagebuch eine zusätzliche Kontextebene hinzu. Indem sie den Ort ihrer Einträge sehen können, können sich Benutzer besser an Erinnerungen und Ereignisse erinnern, die mit diesem bestimmten Ort verbunden sind. Darüber hinaus erleichtert die Möglichkeit, den Standort in Google Maps zu öffnen, den Benutzern die Navigation zu diesem Standort, wenn sie ihn erneut besuchen möchten. Die Standortfunktion in dieser App ist eine nützliche und praktische Funktion, die das Gesamterlebnis beim Führen eines digitalen Tagebuchs verbessern kann.

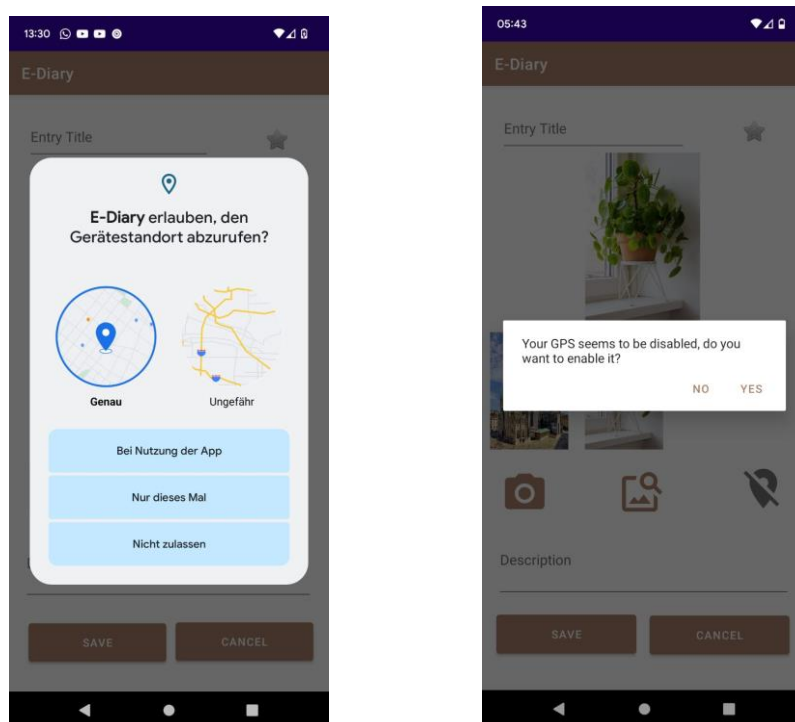


Figure 6 Standort Überprüfung

3.6. Analyse und Kategorisierung mit ML Kit

Die Verwendung von ML Kit in diesem Projekt ist ein Schlüsselaspekt seiner Funktionalität und unterscheidet es von anderen traditionellen Tagebuchanwendungen. ML Kit bietet ein Framework für maschinelle Lernmodelle,

die unter anderem Aufgaben wie Bildkennzeichnung, Texterkennung und Gesichtserkennung ausführen können.

In diesem Projekt wird ML Kit für die Bildkategorisierung verwendet, wodurch die Anwendung Bilder innerhalb von Einträgen automatisch kategorisieren kann. Dies fügt den Einträgen des Benutzers eine Organisations- und Strukturebene hinzu, was die Navigation und das Auffinden bestimmter Informationen erleichtert [9] [10].

Die Integration von ML Kit ermöglicht auch einen schnelleren und effizienteren Bildkategorisierungsprozess, da es lokal auf dem Gerät ausgeführt werden kann, ohne dass eine Internetverbindung erforderlich ist. Dies verbessert die Benutzererfahrung, da nicht mehr darauf gewartet werden muss, dass Bilder von einem externen Server verarbeitet und kategorisiert werden. Die Verwendung von ML Kit unterstreicht den innovativen Ansatz, der beim Design und der Entwicklung dieser Anwendung verfolgt wurde, und macht sie zu einem einzigartigen und wertvollen Werkzeug zum Führen eines digitalen Tagebuchs.

3.7. Listing

Das Abrufen von Informationen und das Anzeigen von Informationen in Listen ist ein wichtiger Aspekt jeder Anwendung, da es den Benutzern ermöglicht, leicht auf ihre Daten zuzugreifen und sie zu verwalten. Dies ist besonders wichtig für eine erweiterte digitale Tagebuchanwendung, bei der Benutzer die Möglichkeit haben, verschiedene Arten von Informationen zu speichern und zu kategorisieren, darunter Bilder, Text und andere Medien.

In diesem Zusammenhang ist die Möglichkeit, Informationen in Listen anzuzeigen und nach verschiedenen Kriterien wie Datum, Bildkategorien und Titel zu filtern, ein wertvolles Feature. Durch den Einsatz von Techniken des maschinellen Lernens wie der Bildkategorisierung kann die Anwendung den Benutzern eine intuitive und effiziente Möglichkeit bieten, ihre Einträge zu sortieren und die gesuchten Informationen zu finden.

Forschungen auf dem Gebiet des Abrufens und Anzeigens von Informationen haben gezeigt, dass Benutzer listenbasierte Schnittstellen anderen Formen der Informationsdarstellung wie Tabellen oder Diagrammen vorziehen. Dies liegt daran, dass Listen einfach zu lesen und zu verstehen sind und es Benutzern ermöglichen, die benötigten Informationen schnell zu scannen und zu identifizieren. [12]

Studien haben auch gezeigt, dass die Möglichkeit, Informationen nach verschiedenen Kriterien wie Datum, Relevanz oder Kategorien zu filtern, die Benutzerfreundlichkeit und Effizienz einer Anwendung erheblich verbessern kann [13]. Insbesondere die Bildkategorisierung und der bildbasierte Informationsabruf

haben in den letzten Jahren große Aufmerksamkeit von Forschern erhalten, wobei zahlreiche Studien verschiedene Techniken zum Klassifizieren und Abrufen von Bildern basierend auf visuellen Merkmalen untersucht haben [14].

Zusammenfassend lässt sich sagen, dass die Fähigkeit, Informationen in Listen anzuzeigen und sie basierend auf verschiedenen Kriterien zu filtern [15], ein kritischer Aspekt einer angereicherten digitalen Tagebuchanwendung ist und Gegenstand umfangreicher Forschung auf dem Gebiet des Abrufens und Anzeigens von Informationen war. Durch den Einsatz von Techniken des maschinellen Lernens, wie die Bildkategorisierung, kann die Anwendung den Benutzern eine intuitive und effiziente Möglichkeit bieten, auf ihre Informationen zuzugreifen und sie zu verwalten.

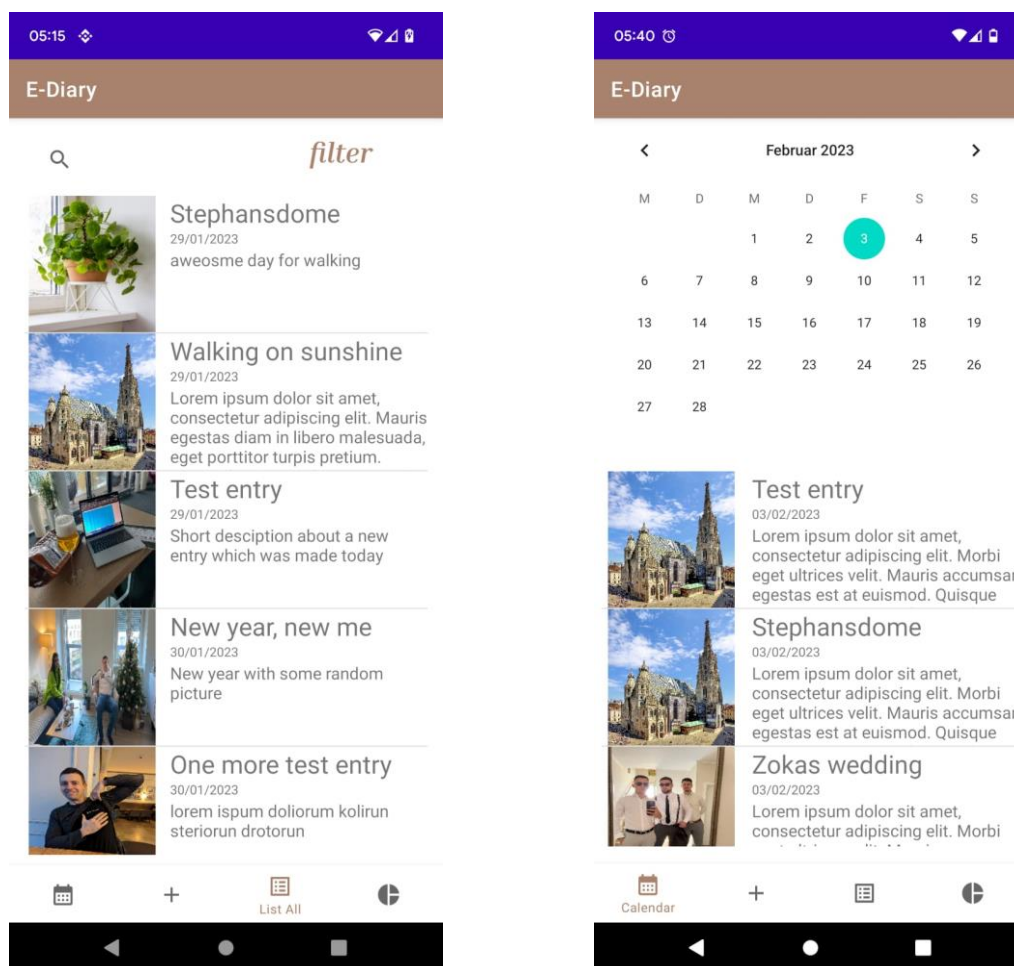


Figure 7. Such- und Anzeigoptionen mit Kalender, Such und Kategorie Filtrierung

4. Implementation

4.1. Register

Der Registrierungsprozess ist ein obligatorischer Schritt, den der Benutzer durchführen muss. Hier muss er ein Konto mit einem Benutzernamen und einem Passwort einrichten, damit er zur Anmeldeseite gelangen kann. Ohne sie kann er sich nicht anmelden und die Anwendung nicht verwenden. Der Benutzer muss dasselbe Passwort zweimal eingeben, um sicherzustellen, dass der Benutzer das gewünschte Passwort richtig eingegeben hat. Die Anwendung vergleicht die beiden Eingaben und bestätigt, dass das gewünschte Passwort korrekt eingegeben wurde, oder zeigt ein Popup an, um sicherzustellen, dass die Passwörter korrekt sind.

Nach Eingabe der korrekten Anmeldedaten werden die Daten an den UserController.java weitergeleitet, der über den DatabaseController.java Zugriff auf die Datenbank hat. Der DatabaseController speichert die Daten in der Datenbank für späteren Authentifizierungen.

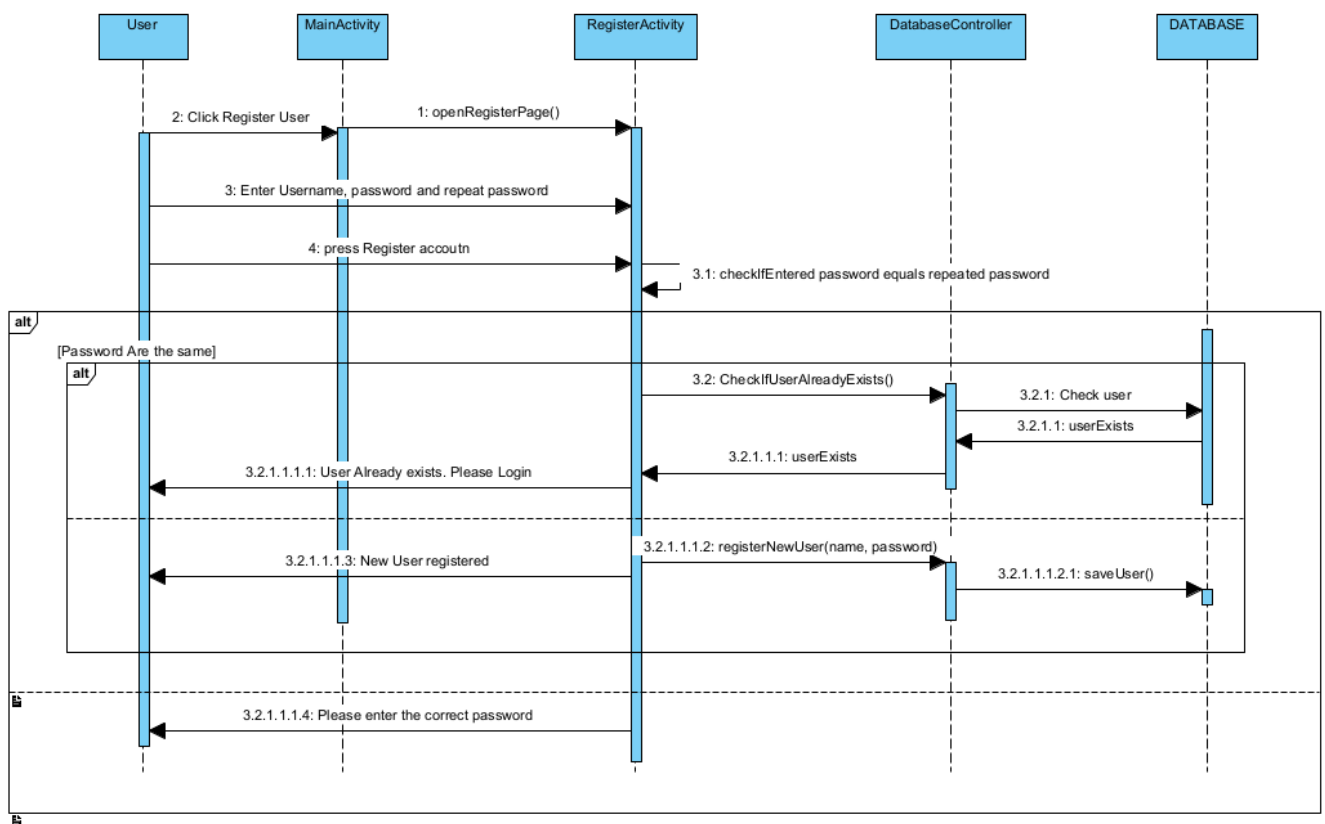


Figure 8 Sequence Diagram of the Register process

4.2. Log in

Damit sich der Benutzer anmelden und die Anwendung nutzen kann, muss er ein Benutzerkonto erstellen. Falls er kein Konto erstellt und zur Anmeldeseite geht, sind beide Optionen, Fingerabdruck- und Kennwortanmeldung, nicht verfügbar.

Die erste Möglichkeit, sich in das Konto einzuloggen, ist die Passwortoption, bei der der Benutzer das Passwort aus der Registrierung in das Eingabefeld eingibt und auf die Schaltfläche Login mit Passwort drückt.

Danach wird das Passwort abgerufen und mit dem gespeicherten Passwort in der Datenbank verglichen. Dies geschieht im UserController. Bei korrekter Passworтеingabe wird der Benutzer auf die DisplayPage mit der Kalenderansicht weitergeleitet.

Wenn ein falsches Passwort eingegeben wird, wird eine Popup-Meldung angezeigt, um das richtige Passwort einzugeben.

Die zweite Möglichkeit, sich anzumelden, ist mit dem biometrischen Fingerabdruckscanner, wenn das Gerät dies unterstützt. Wenn das Scannen des Fingerabdrucks erfolgreich ist, wird der Benutzer zum gleichen Seitenaufruf weitergeleitet wie im ersten Fall.

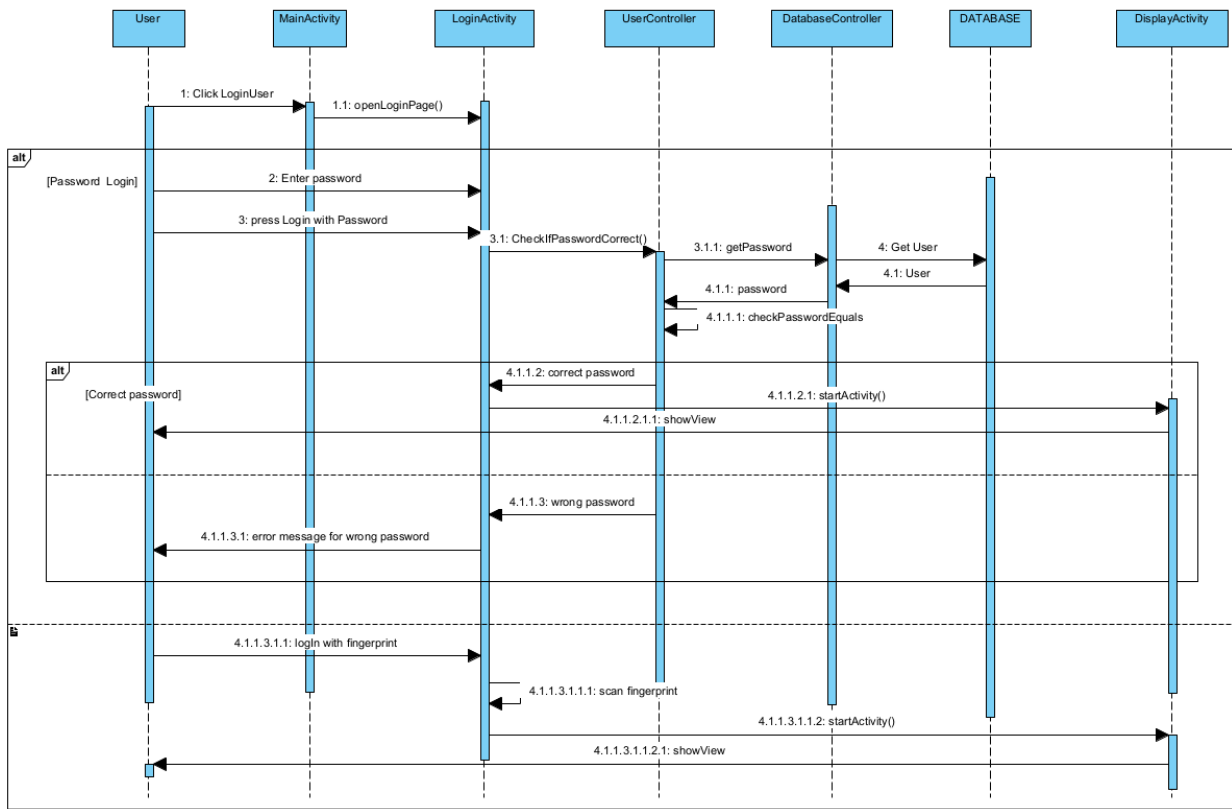


Figure 9 Sequence Diagram of the login process

4.3. Entry erstellungs process

Hier gehen wir näher auf den Erstellungsprozess des Eintrags ein.

Wenn der Benutzer ein Bild aus der Galerie oder mit der Kamera aufnimmt, wird das in Bitmap konvertierte Bild in einer galleryImages Liste von Bitmaps gespeichert. Die Bilder aus dieser Liste werden in der horizontalen angezeigt mit einer RecyclerView . Die RecyclerView verwendet den RecycleGalleryAdapter. Da der Adapter über einen onItemClick-Listener verfügt, wird beim Klicken auf Elemente in der Galerie das Bild aus der Galerie von der angeklickten Position angezeigt.

Wenn der Benutzer alle Daten eingibt auf der Erstellungsseite und auf Speichern drückt, als ersters wird es geprüft, ob alle Daten eingegeben wurden. Falls nicht, erscheint eine Fehlermeldung, ansonsten werden die Daten an den EntryController weitergeleitet, wie wir im Bild sehen können, wo sie dann zu der DatabaseController.

```
public void save(View view) throws ParseException, IOException {  
    if (title.getText().toString().equals("") || title.getText().length() <= 0  
        || description.getText().toString().equals("") || description.getText().length() <= 0  
        || galleryImages.isEmpty()) {  
        Toast.makeText(context, this, "Please fill up the fields and at least one picture picture.", Toast.LENGTH_SHORT).show();  
        return;  
    }  
  
    entryController.saveEntry(title.getText().toString(), description.getText().toString(), latitude, longitude, isFavorite, galleryImages);  
  
    Toast.makeText(context, this, "NEW ENTRY CREATED", Toast.LENGTH_SHORT).show();  
  
    initializeAndRefreshData();  
}
```

Figure 10 Entry daten Weiterleitung für erstellung

Im EntryController wird ein Entry-Objekt mit den bereitgestellten Daten erstellt und für jedes Bild aus der Galerieliste die Analyse in separaten Threads gestartet, die in 4.5 näher beschrieben werden. Das Eintragsobjekt wird dann an den DatabaseController übergeben, wo es in der Datenbank gespeichert wird.

```
public void saveEntry(String title, String description, double latitude, double longitude, boolean isFavorite, List<Bitmap> galleryImages) {  
    Entry entry = new Entry(title, description, latitude, longitude, isFavorite);  
    for (Bitmap image : galleryImages) {  
        ((Runnable) () -> ImageAnalyzer.runClassification(context, entry.getUID(), Converter.resizeBitmap(image))).run();  
    }  
  
    databaseController.addNewEntry(entry);  
}
```

Figure 11 Initialisierung des EntryImage Analyseprozesses

4.4. Database

Die gesamte Kommunikation zum Speichern, Abrufen, Aktualisieren und Löschen von Daten in der Datenbank erfolgt über den DatabaseController, der den SQLiteOpenHelper erweitert. Nach der Initialisierung des Controllers werden alle drei Tabellen in der Methode onCreate angelegt. Eine davon ist die bereits erwähnte User-Tabelle und die anderen beiden, eng miteinander verknüpft, sind die entry-Tabelle und die entryImage-Tabelle. In der Eintragstabelle speichern wir die grundlegenden Informationen des Eintrags wie Titel, Datum, Beschreibung und andere [mehr in Abbildung 3.neso]. In der entryImage-Tabelle werden alle Bilder mit den analysierten Daten gespeichert, die dem richtigen Eintrag über die Eintrags-UID zugeordnet sind.

Unten sehen wir ein Beispiel für jede der CRUD-Operationen.

Das CREATE bezieht sich auf das Speichern von Entry Daten in die Datenbank. Durch das `this.getWritableDatabase()`, bezieht sich auf dem DatabaseController, bekommen wir Zugriff auf die Datenbank zum Schreiben der Daten. Danach legen wir die Objektdaten in die entsprechende Tabellenspalte und fügen diese in die Tabelle ein.

```
public void addNewEntry(Entry entry) {  
  
    DATABASE = this.getWritableDatabase();  
  
    ContentValues values = new ContentValues();  
    values.put(UID, entry.getUID());  
    values.put(DATE, entry.getCreationDate().toString());  
    values.put(TITLE, entry.getTitle());  
    values.put(FAVORITE, entry.isFavorite());  
    values.put(DESCRIPTION, entry.getDescription());  
    values.put(LATITUDE, Double.toString(entry.getLatitude()));  
    values.put(LONGITUDE, Double.toString(entry.getLongitude()));  
  
    DATABASE.insert(ENTRY_TABLE, nullColumnHack, null, values);  
    DATABASE.close();  
}
```

Figure 12. Hinzufügen von Daten zur Datenbank

Beim READ ist es so, dass wir Daten aus der Datenbank lesen, indem wir einen Cursor mit einer einfachen SQL-Abfrage einrichten, der alle Daten entsprechend der Abfrage abrufen.

```

private List<EntryImage> getImagesByEntryId(String uid) {

    Cursor entries = DATABASE.rawQuery("SELECT * FROM " + CATEGORY_TABLE + " WHERE ENTRY_ID = " + uid + " ", selectionArgs: null);

    List<EntryImage> entryImages = new ArrayList<>();

    if (entries.moveToFirst()) {
        do {
            EntryImage entryImage = new EntryImage(entries.getString(0), Float.valueOf(entries.getString(1)), entries.getString(2));
            entryImages.add(entryImage);
        } while (entries.moveToNext());
    }
    entries.close();

    return entryImages;
}

```

Figure 13. Abrufen von Daten aus der Datenbank

Der UPDATE Prozess ist unkompliziert, wobei wir die vorhandenen, upgedateten Werte eines Objekts mithilfe der integrierten update Funktion in der entsprechenden Tabelle ändern können. Durch die Eingabe der neuen Werte werden diese korrekt den richtigen Zeilen zugeordnet.

```

public void updateFavorite(Entry entry){
    DATABASE = this.getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(UID, entry.getUID());
    values.put(ENTRY_ID, entry.getCreationDate());
    values.put(TITLE, entry.getTitle());
    values.put(FAVORITE, entry.isFavorite());
    values.put(DESCRIPTION, entry.getDescription());
    values.put(LATITUDE, Double.toString(entry.getLatitude()));
    values.put(LONGITUDE, Double.toString(entry.getLongitude()));

    DATABASE.update(ENTRY_TABLE, values, whereClause: UID + " = ?", new String[]{entry.getUID()});
}

```

Figure 14. Aktualisieren von Daten in die Datenbank

Schließlich ermöglicht uns die DELETE-Operation, Objekte zu entfernen, indem wir die eingebaute "delete" Funktion verwenden und die entsprechenden IDs der zu löschenden Objekte übergeben.

```

public void deleteEntryById(String uid) {

    DATABASE = this.getWritableDatabase();
    DATABASE.delete(ENTRY_TABLE, whereClause: UID + "=?", new String[]{uid});
    DATABASE.delete(CATEGORY_TABLE, whereClause: ENTRY_ID + "=?", new String[]{uid});
    DATABASE.close();
}

```

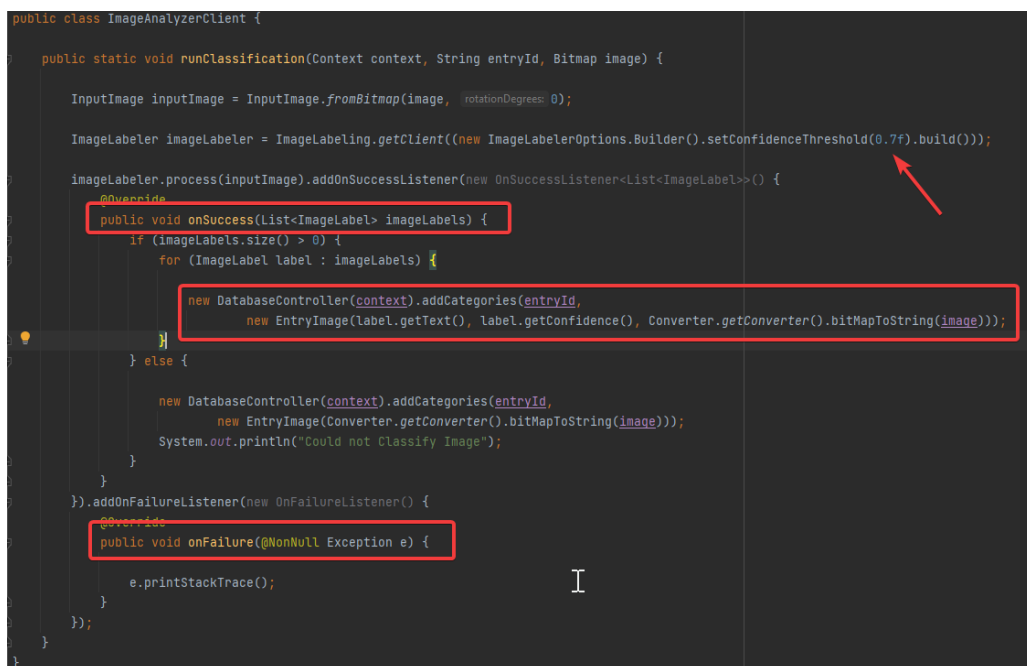
Figure 15. Löschen von Daten aus der Datenbank

4.5. Image Analyzer – MLKit

Die Kategorisierung der Bilder in der angereicherten Tagebuch-App erfolgt automatisch mit MLKit, einem lokalen Framework für maschinelles Lernen auf dem Gerät [16]. Dadurch kann die App Bilder sofort kategorisieren, ohne auf eine Internetverbindung angewiesen zu sein. Der Kategorisierungsprozess wird jedes Mal ausgelöst, für jedes Bild das zu der Galerie hinzugefügt würde, sodass Einträge basierend auf dem kategorisierten Bildinhalt sortiert und gefiltert werden können. Durch die Verwendung von MLKit bietet die App eine schnelle und effiziente Möglichkeit, Bilder zu kategorisieren, was die Organisation und Verwaltung der Tagebucheinträge erleichtert. Diese Funktion verbessert die allgemeine Benutzererfahrung, indem sie eine schnelle und einfache Navigation und den Abruf von Tagebucheinträgen ermöglicht.

Um die ML-Kit-API zu verwenden, müssen wir die zwei dependencies: *implementation 'com.google.mlkit:image-labeling:17.0.7'* und *implementation 'com.google.mlkit:image-labeling-common:18.1.0'* zu unserem build.gradle hinzufügen.

Wenn der Benutzer einen Eintrag erstellen möchte, wie es in Eintrag erstellen / Eintragsmodell, wird die Funktion `runClassification()` aus der Klasse `ImageAnalyzerClient` für jedes Bild einzeln in einem separaten Thread aufgerufen. In der Methode `runClassification()` richtet die ML-Kit API einen `ImageLabeler`-Client ein, der den Prozess über das `Bitmap`-Bildobjekt ausführt. Durch das Festlegen von `setConfidenceThreshold()` begrenzen wir die Antwortergebnisse, die bestätigt werden, dass sie über diesem Grenzwert liegen. Nach erfolgreicher Kategorisierung werden für jede zurückgegebene Kategorie `EntryImages` erstellt und in der Datenbank gespeichert. Wenn wir keine Kategorie für das Bild erhalten, wird ein `EntryImage` mit der Standardkategorie "NO_CATEGORY" erstellt. und gespeichert.



```
public class ImageAnalyzerClient {

    public static void runClassification(Context context, String entryId, Bitmap image) {

        InputImage inputImage = InputImage.fromBitmap(image, rotationDegrees: 0);

        ImageLabeler imageLabeler = ImageLabeling.getClient((new ImageLabelerOptions.Builder().setConfidenceThreshold(0.7f).build()));

        imageLabeler.process(inputImage).addOnSuccessListener(new OnSuccessListener<List<ImageLabel>>() {

            @Override
            public void onSuccess(List<ImageLabel> imageLabels) {
                if (imageLabels.size() > 0) {
                    for (ImageLabel label : imageLabels) {
                        new DatabaseController(context).addCategories(entryId,
                            new EntryImage(label.getText(), label.getConfidence(), Converter.getConverter().bitMapToString(image)));
                    }
                } else {
                    new DatabaseController(context).addCategories(entryId,
                        new EntryImage(Converter.getConverter().bitMapToString(image)));
                    System.out.println("Could not Classify Image");
                }
            }
        }).addOnFailureListener(new OnFailureListener() {

            @Override
            public void onFailure(@NonNull Exception e) {
                e.printStackTrace();
            }
        });
    }
}
```

Figure 17. MLKit Klienten Aufruf für Bildanalyse und Bearbeitung der reponse Daten

4.6. Listing

4.6.1. Kalender Listing

Zum Anzeigen von Einträgen verwenden wir den EntryListAdapter, der den ArrayAdapter<?> erweitert, an den die Liste der anzuzeigenden Einträge übergeben wurde.

```
public class EntryListAdapter extends ArrayAdapter<Entry> {  
  
    private Context context;  
    private int resource;  
  
    public EntryListAdapter(@NonNull Context context, int resource, List<Entry> entries) {  
        super(context, resource, entries);  
  
        this.context = context;  
        this.resource = resource;  
    }  
  
    @SuppressWarnings("ViewHolder")  
    @NonNull  
    @Override  
    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
  
        LayoutInflater inflater = LayoutInflater.from(context);  
        convertView = inflater.inflate(resource, parent, false);  
  
        TextView titleTextView = (TextView) convertView.findViewById(R.id.textView);  
        ImageView imageView = (ImageView) convertView.findViewById(R.id.imageView);  
        TextView descriptionTextView = (TextView) convertView.findViewById(R.id.textView2);  
        TextView dateView = convertView.findViewById(R.id.list_date);  
  
        titleTextView.setText(getItem(position).getTitle());  
        imageView.setImageBitmap(Converter.stringToBitmap(getItem(position).getFirstImage()));  
        descriptionTextView.setText(getItem(position).getDescription());  
        dateView.setText(getItem(position).getCreationDate());  
  
        return convertView;  
    }  
}
```

Figure 18. Horizontal List Adapter

Wenn auf ein Datum in der Kalenderansicht geklickt wird [Figure 19], wird die Listenansicht aktualisiert basierend auf der Liste, die von der EntryController-Funktion getEntriesByDate zurückgegeben wird [Figure 20].

```
private void setOnCalendarClick() {  
  
    calendarView.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {  
        @Override  
        public void onSelectedDayChange(@NonNull CalendarView calendarView, int year, int month, int day) {  
  
            updateListView(entryController.getEntriesByDate(day, month: month + 1, year));  
        }  
    });  
}
```

Figure 19. Funktionen Aufruf für Datum Filterung

```

40 public List<Entry> getEntriesByDate(int day, int month, int year) {
41
42     String date;
43
44     String monthString = month < 10 ? "0" + month : String.valueOf(month);
45     String dayString = day < 10 ? "0" + day : String.valueOf(day);
46
47     date = dayString + "/" + monthString + "/" + year;
48
49     List<Entry> dateEntries = new ArrayList<>();
50
51     for (Entry entry : databaseController.getAllEntries()) {
52
53         if (entry.getCreationDate().equals(date)) {
54             dateEntries.add(entry);
55         }
56     }
57
58     return dateEntries;
59 }

```

Figure 20. Entry Filterung bei Datum

4.6.2. Kategorie Filtrierung

Das Filtern von Einträgen in dieser Anwendung funktioniert ähnlich wie das Auflisten von Einträgen im Kalender. Die gefilterte Liste wird an den Adapter übergeben, indem die Funktion `getEntriesByFilteredCategories()` vom `EntryController` aufgerufen wird. Diese Funktion ruft alle Einträge ab, die einer Kategorie zugeordnet sind, die zuvor aus dem Popup-Diagramm ausgewählt wurde. Das Ergebnis dieser Operation ist eine gefilterte Liste von Einträgen, die dann dem Benutzer durch die Verwendung des Adapters angezeigt wird.

```

107 public List<Entry> getEntriesByFilteredCategories(ArrayList<String> selectedCategories) {
108
109     List<Entry> filteredEntries = new ArrayList<>();
110
111     for (Entry entry : getAllEntries()) {
112         for (EntryImage image : entry.getImages()) {
113             if (selectedCategories.contains(image.getCategory())
114                 || (selectedCategories.contains("Favorite") && entry.isFavorite())) {
115
116                 filteredEntries.add(entry);
117                 break;
118             }
119         }
120     }
121
122     return filteredEntries;
123 }
124

```

Figure 21. Entry Filterung nach ausgewählten Kategorien

4.7. Diagram

Die MPAndroidChart-Bibliothek ist eine externe Abhängigkeit, die in der Anwendung verwendet wird, um Daten in Form eines Kreisdiagramms visuell darzustellen. Das Diagramm zeigt die Verteilung der Einträge auf verschiedene Kategorien und zeigt den Prozentsatz der Einträge pro Kategorie. Durch die Verwendung dieser Bibliothek ist die Anwendung in der Lage, Benutzern eine einfache und intuitive Möglichkeit zu bieten, die Verteilung ihrer Einträge anzuzeigen, was ihnen hilft, Einblicke zu gewinnen und ihre Daten besser zu verstehen. Diese Funktion ist ein wichtiger Aspekt der Anwendung, da sie es den Benutzern ermöglicht, ihre Eingaben schnell und einfach zu analysieren, was es ihnen erleichtert, Trends, Muster und Verbesserungsmöglichkeiten zu erkennen. Mit MPAndroidChart bietet die Anwendung eine flexible und anpassbare Lösung zur benutzerfreundlichen Darstellung und Analyse von Daten.

```
50 private void setupPieChart() {
51     pieChart.setDrawHoleEnabled(true);
52     pieChart.setUsePercentValues(true);
53     pieChart.setEntryLabelTextSize(12);
54     pieChart.setEntryLabelColor(Color.BLACK);
55     pieChart.setCenterText("Entry by Categories");
56     pieChart.setCenterTextSize(24);
57     pieChart.getDescription().setEnabled(false);
58
59     Legend l = pieChart.getLegend();
60     l.setVerticalAlignment(Legend.LegendVerticalAlignment.TOP);
61     l.setHorizontalAlignment(Legend.LegendHorizontalAlignment.RIGHT);
62     l.setOrientation(Legend.LegendOrientation.VERTICAL);
63     l.setDrawInside(false);
64     l.setEnabled(true);
65 }
```

Figure 22. Diagramm einrichtung

```
68 private void loadPieChartData() {
69
70     ArrayList<PieEntry> entries = new ArrayList<>();
71     for (Map.Entry<String, Integer> set : new EntryController(context, ChartActivity.this).categoryPercentage().entrySet())
72         entries.add(new PieEntry(set.getValue(), set.getKey()));
73
74     ArrayList<Integer> colors = new ArrayList<>();
75
76     for (int color : ColorTemplate.MATERIAL_COLORS)
77         colors.add(color);
78
79     for (int color : ColorTemplate.VORDIPLOM_COLORS)
80         colors.add(color);
81
82     PieDataSet dataSet = new PieDataSet(entries, "Entry by Categories");
83     dataSet.setColors(colors);
84
85     PieData data = new PieData(dataSet);
86     data.setDrawValues(true);
87     data.setValueFormatter(new PercentFormatter(pieChart));
88     data.setValueTextSize(12f);
89     data.setValueTextColor(Color.BLACK);
90
91     pieChart.setData(data);
92     pieChart.invalidate();
93
94     pieChart.animateY(durationMillis: 1400, Easing.EaseInOutQuad);
95 }
```

Figure 23. Laden von Daten in das Diagramm

4.8 Technology stack

IDE	Android Studio
Programmiere Sprache	Java
Database	SQLite
UML-Tool	Visual Paradigm
Markup Language	XML

5. Future Work

In Zukunft kann es sich lohnen, einen Wechsel von MLKit zu Google Vision AI für die Bildanalyse in einer digital angereicherten Tagebuchanwendung in Erwägung zu ziehen. Google Vision AI [16] ist eine leistungsstarke Bildanalyseplattform, die eine Reihe von Funktionen bietet, darunter Objekterkennung, Stimmungsanalyse und Bildkennzeichnung. Durch die Integration von Google Vision AI mit DBpedia, einem großen Knowledge Graph aus verknüpften Daten, kann die Anwendung die enorme Menge an Informationen nutzen, die in DBpedia [17] verfügbar sind, um dem Benutzer noch aussagekräftigere Einblicke und Kontext zu bieten.

Die Integration von Google Vision AI und DBpedia kann der Anwendung auch mehrere Vorteile bringen. Beispielsweise kann Google Vision AI eine genauere Bildkennzeichnung bereitstellen, während DBpedia eine Fülle von Kontextinformationen zu den Objekten und Szenen in den Bildern bereitstellen kann. Diese Informationen können verwendet werden, um das Tagebucherlebnis zu verbessern, indem ein vollständigeres Bild der Erfahrungen des Benutzers bereitgestellt wird, einschließlich der Zeit, des Ortes und des Wetters zum Zeitpunkt der Aufnahme des Bildes.

Darüber hinaus kann die Integration von Google Vision AI und DBpedia auch neue Möglichkeiten für die Anwendung eröffnen, z. B. indem es Benutzern ermöglicht wird, verwandte Informationen und Kenntnisse über die Objekte und Szenen in ihren Bildern zu erkunden. Dies kann eine ansprechendere und informativere Erfahrung für den Benutzer schaffen und kann die fortgesetzte Verwendung der Anwendung fördern.

Insgesamt hat die Integration von Google Vision AI und DBpedia das Potenzial, die Fähigkeiten und den Wert einer digital angereicherten Tagebuchanwendung erheblich zu steigern. Durch die Nutzung der neuesten Technologie und großer Mengen verknüpfter Daten kann die Anwendung dem Benutzer ein reichhaltiges und personalisiertes Erlebnis bieten und ihm helfen, seine Erfahrungen auf sinnvolle und ansprechende Weise zu reflektieren.

Neben dem Wechsel von MLKit zu Google Vision AI und der Integration mit DBpedia kann auch die Unterstützung verschiedener Medientypen wie Audio und Video als zukünftige Richtung für die Anwendung für digital angereicherte Tagebücher in Betracht gezogen werden.

Audio und Video bieten dem Benutzer zusätzliche Möglichkeiten, seine Erfahrungen festzuhalten und zu reflektieren, und sie können im Vergleich zu Text und Bildern allein ein immersiveres und ansprechenderes Erlebnis bieten. Durch Hinzufügen der Unterstützung für diese Medientypen kann die Anwendung ein vollständigeres Bild der Erfahrungen des Benutzers liefern und ihm helfen, seine Erinnerungen im Laufe der Zeit besser zu reflektieren.

Darüber hinaus kann die Integration von Audio und Video in die Anwendung auch neue Möglichkeiten für die Analyse und Generierung von Erkenntnissen eröffnen. Beispielsweise können Spracherkennungsalgorithmen verwendet werden, um Audioaufnahmen automatisch zu transkribieren, und Stimmungsanalysealgorithmen können verwendet werden, um Emotionen und Stimmungen sowohl in Audio- als auch in Videoinhalten zu identifizieren. Diese Informationen können verwendet werden, um das Tagebuchelebnis zu verbessern, indem ein vollständigeres Bild des emotionalen Zustands des Benutzers zum Zeitpunkt der Aufzeichnung des Inhalts bereitgestellt wird.

Zusammenfassend lässt sich sagen, dass das Hinzufügen von Unterstützung für verschiedene Medientypen, wie Audio und Video, der Anwendung für digital angereicherte Tagebücher ein neues Maß an Reichhaltigkeit und Engagement verleihen kann. Durch die Nutzung der neuesten Technologie und modernster Algorithmen kann die Anwendung dem Benutzer ein personalisiertes, immersives und wertvolles Erlebnis bieten und ihm helfen, seine Erfahrungen auf sinnvolle und ansprechende Weise zu reflektieren.

6. Schlussfolgerung

Zusammenfassend zielte das Projekt darauf ab, eine erweiterte digitale Tagebuch-App zu erstellen, die es den Benutzern ermöglicht, ihre täglichen Gedanken, Erfahrungen und Momente auf einfache und organisierte Weise zu verfolgen. Die App nutzt ML Kit um Bilder und Texteingaben zu kategorisieren, sodass Benutzer ihre Einträge basierend auf verschiedenen Kriterien wie Datum, Ort und Kategorien filtern und anzeigen können. Darüber hinaus verfügt die App auch über eine Funktion zur Anzeige der Einträge in einem Tortendiagrammformat, das die Verteilung der Einträge pro Kategorie darstellt.

Das Projekt bot eine wertvolle Lernerfahrung, den Erwerb praktischer Kenntnisse in der Entwicklung von Android-Anwendungen, der Integration von KI- und maschinellen Lerntechnologien und der Arbeit mit Datenbanken. Aufgrund von Änderungen in der Technologielandschaft wurden jedoch einige geplante Funktionalitäten nicht implementiert, wie z. B. die Integration von DBPedia mit Cloud Vision AI und die Implementierung fortschrittlicherer Textanalysetechniken. Trotz dieser Einschränkungen dient die App als starke Grundlage für die zukünftige Entwicklung und hat das Potenzial, ein beliebtes Tool für das persönliche Diaring und das Verwalten von Erinnerungen zu werden.

7. Biography

- [1] L. D. Ross and K. L. Miller "The Use of Mobile Technology for Mental Health: A Systematic Review", 2015.
- [2] W. Pieterse, J. J. J. M. Roks, and L. Markopoulos: "The impact of an enriched digital diary on well-being: A randomized controlled trial".
- [3] „ Pen Heaven Survey Results “. [Online]. <https://www.penheaven.co.uk/blog/diary-survey-results>
- [4] J. P. BOC, L. T. Nguyen, and C. S. J. Bakir: "Mobile application privacy: A systematic review", 2017.
- [5] J. Jain, A. Ross, and S. Nandakumar: "Usability and security in biometric authentication systems", 2016.
- [6] M.H. Kerguelen, T. Défréville and P.B. Godefroid: "SQLite for mobile devices: Where does it stand?" 2009
- [7] Nikola Obradovic, Aleksandar Kelec, Igor Dujlovic: "Performance analysis on Android SQLite database", 2019.
- [8] G.H. Meiklejohn and D.S.J. van der Walt: "SQLite for Mobile Devices: A Survey".
- [9] "Deep Learning for Personal Memory Management" by D. Lian, X. Zhang, and Y. Tao
- [10] L. N. Al-Rashidi, N. R. Mohd, and M. A. Karim: "Mobile Personal Memory Management using Deep Learning".
- [11] M. Li, Y. Li, and X. Wang: "Context-aware Image Classification for Personal Diaries".
- [12] "The Effects of Task Complexity and Display Type on List Search Performance" by Shneiderman, 1983.
- [13] Pirolli and Card: "Information Foraging Theory: Adaptive Interaction with Information", 1999.
- [14] "Image Classification in Personal Memory Management: A Comparative Study" by X. Zhang, Y. Tao, and D. Lian
- [15] "Filtering Techniques in User Interfaces: A Review" by Raab and Buchanan, 2000.
- [16] "Google Cloud Vision API" [Online]. Available: <https://cloud.google.com/vision>.
- [17] "DBpedia" [Online]. Available: <https://www.dbpedia.org/>