

Architektur und Design

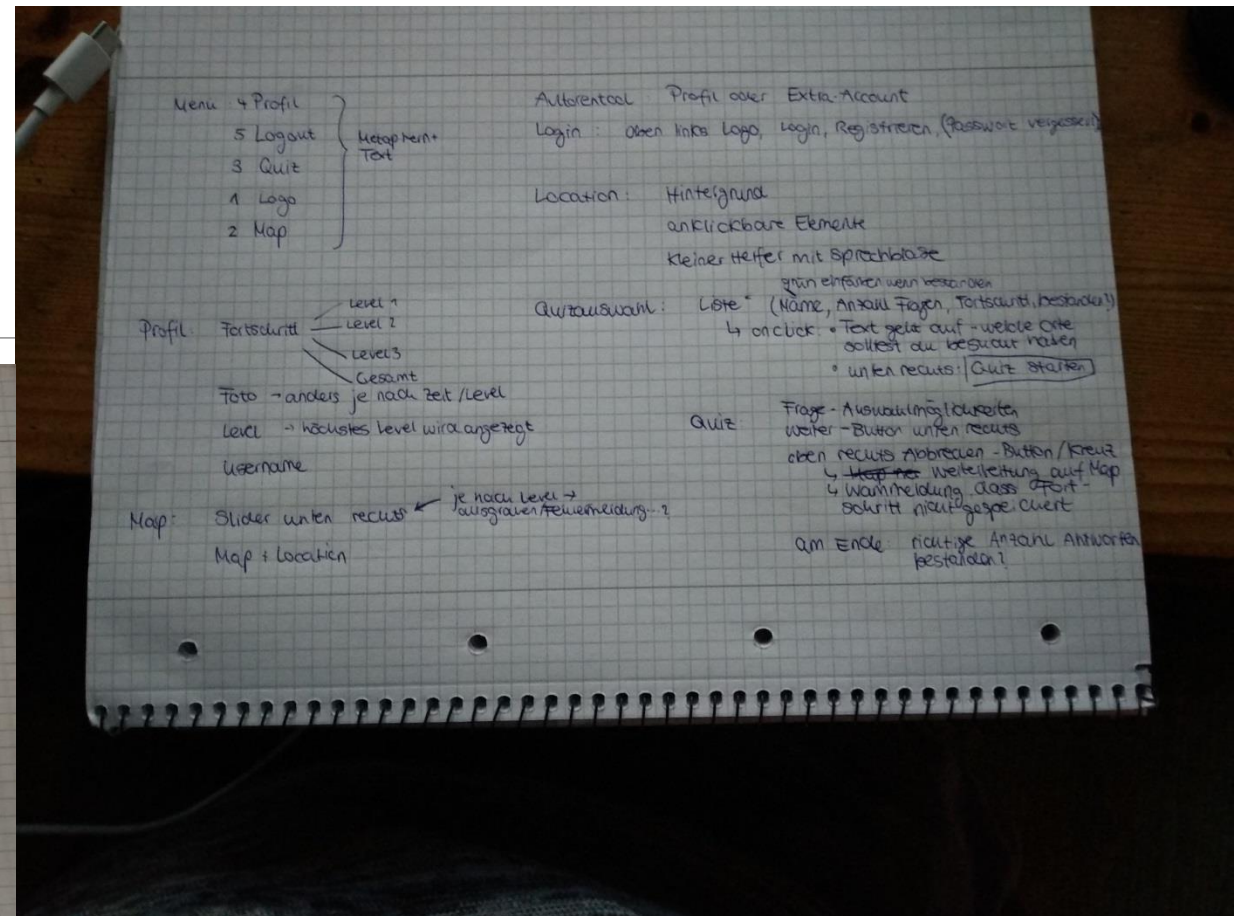
DESIGNPROZESS UND HEURISTIKEN

Inhalt

1. Prototypen
2. Sitemap
3. Datenbankmodell
4. Design und Umsetzung
5. Heuristiken

1. Prototypen

Papier Prototypen



Zu Beginn erarbeiteten wir gemeinsam die benötigten Bestandteile des Programms, die wichtigsten Seiten und die sich darauf befindenden Elemente und erstellten Prototypen, zuerst auf Papier und anschließend mit Axure.

Login Prototyp

Von der Loginseite aus sollten Benutzer auf eine Seite zur Accounterstellung kommen und sich einloggen können, wenn sie einen Account besitzen.
Es sollte außerdem im Hintergrund ein Bild von Würzburg zu sehen sein, vorzugsweise mit einigen der Sehenswürdigkeiten / späteren Spielorten.



Map Prototyp

Auf der Map sollte es anklickbare Punkte, bei uns als Locations bezeichnet, geben. Verschiedene Orte sollten durch verschiedene Icons symbolisiert werden, wie es auf dem Map Prototyp zu sehen ist.

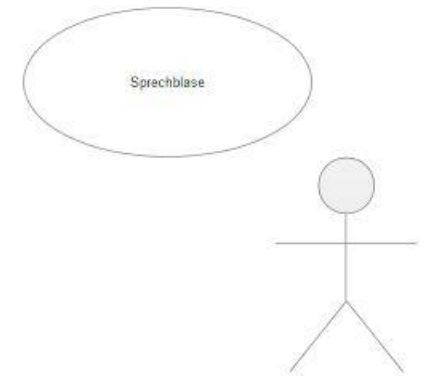


Location Prototyp

Wenn der Benutzer auf eine Location klickt, sollte er auf die Seite dieser Location weitergeleitet werden. Dort sollten dem Benutzer Informationen zu der Location angezeigt werden. Eine Figur übernimmt die Rolle des „Stadtführers“ und vermittelt den Benutzern einige Kerninformationen zu dem angeklickten Ort. Das Ganze sollte mit einem Bild der Location im Hintergrund geschehen.

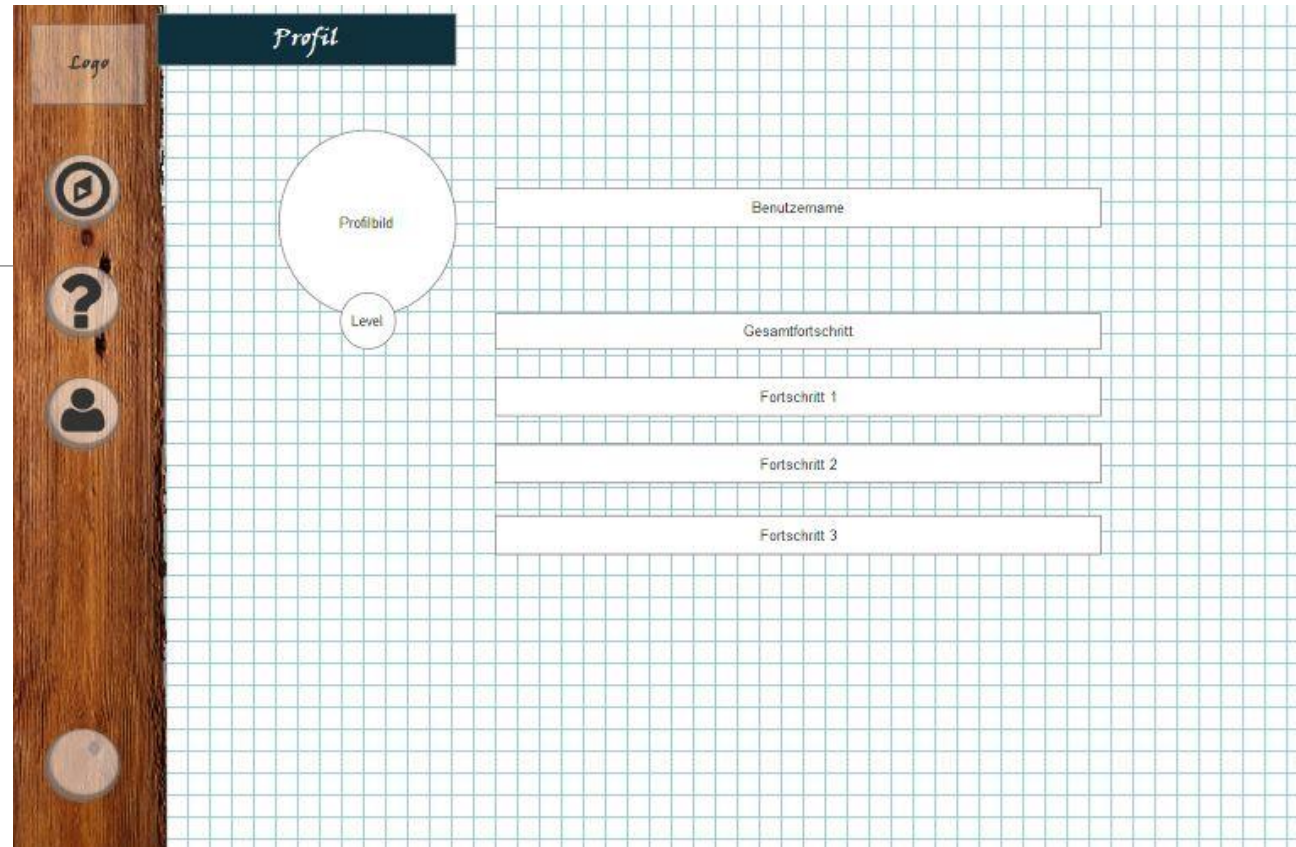


Bild
der Location



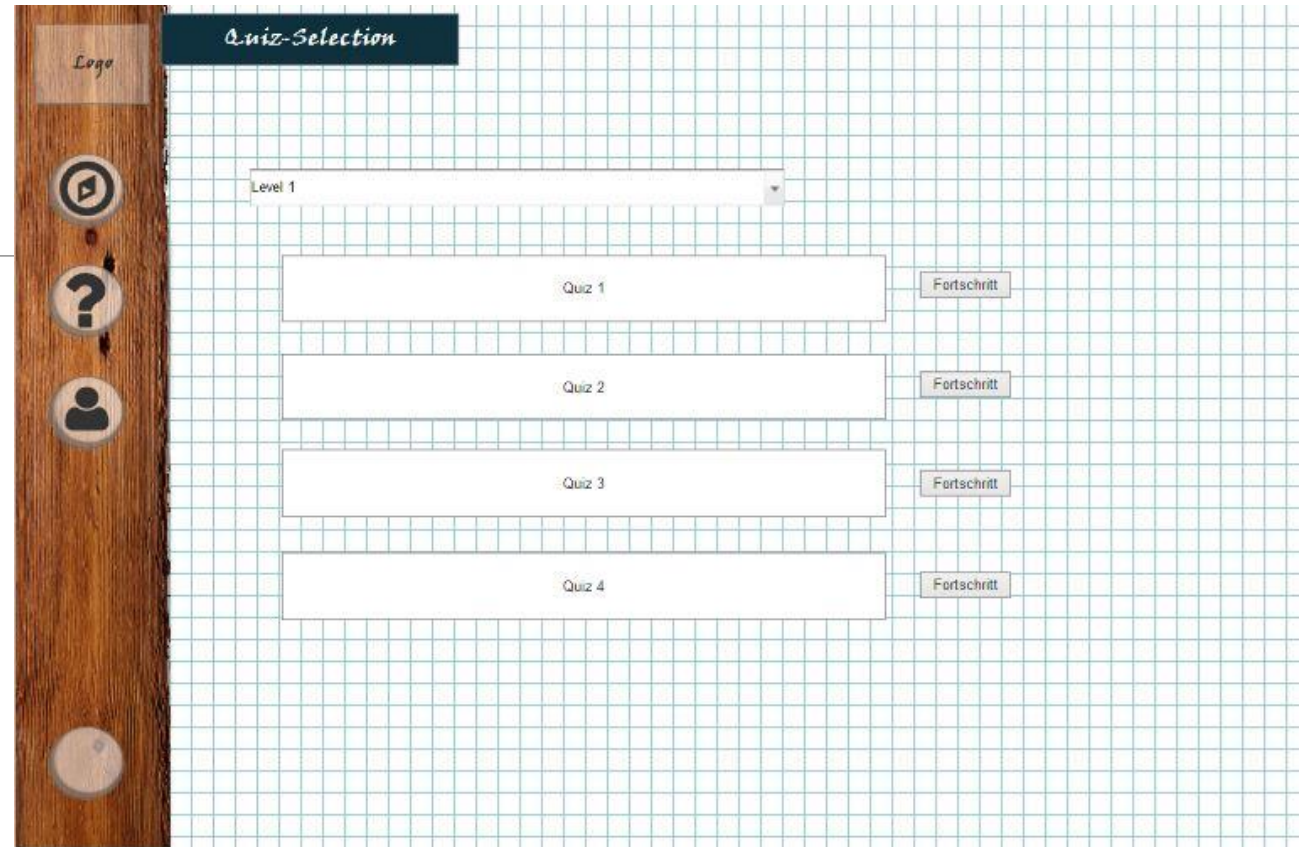
Profil Prototyp

Auf der Profilseite sollte der Benutzer seine Daten sehen können, dazu gehören sein derzeitiges Level, sein Name und Fortschritt im Spiel.



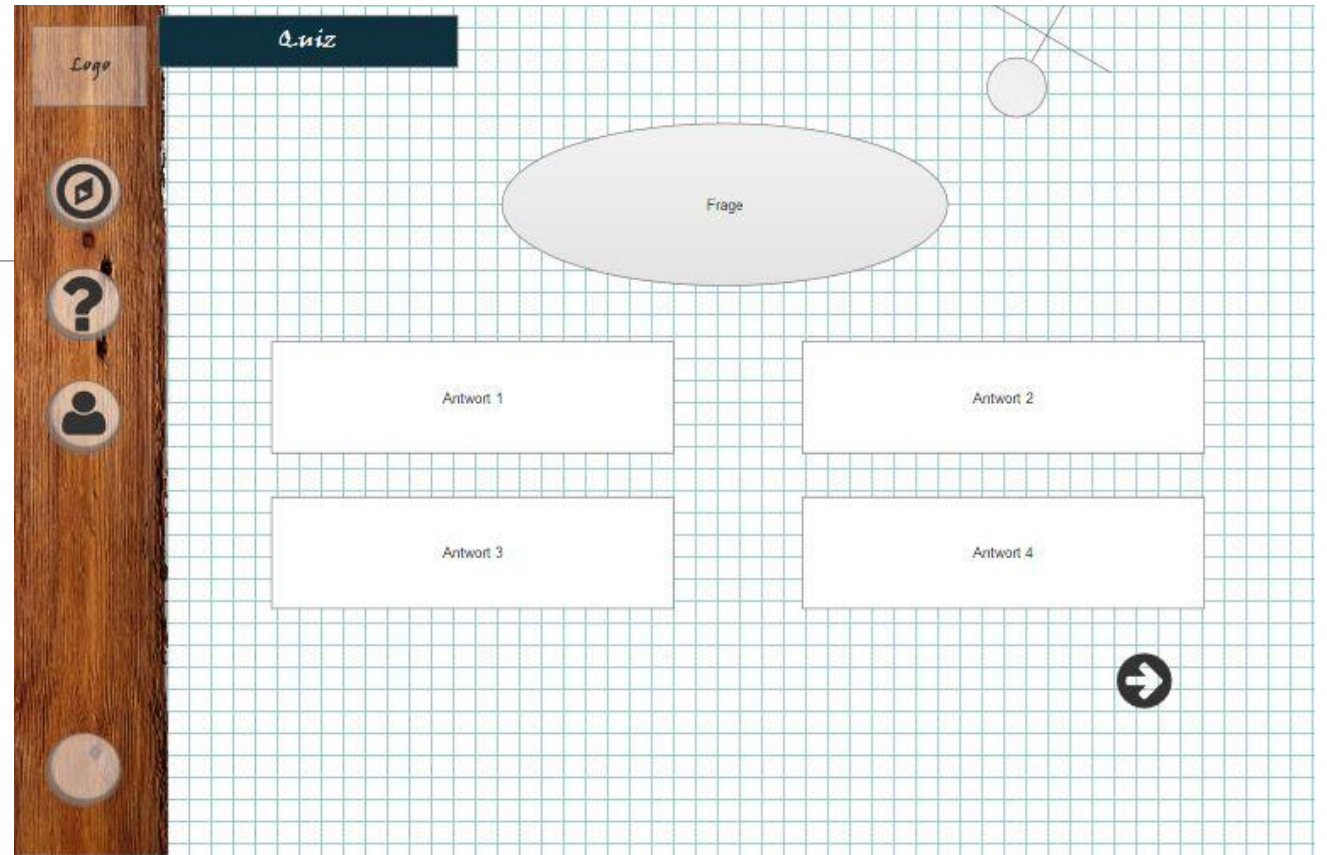
Quiz Prototyp

In der Quizauswahl sollte ein Benutzer wählen können, welches der verfügbaren Quizzes er bearbeiten will. Außerdem sollte dem Benutzer sein Fortschritt bzw. Bestleistung in jedem Quiz angezeigt werden.



Question Prototyp

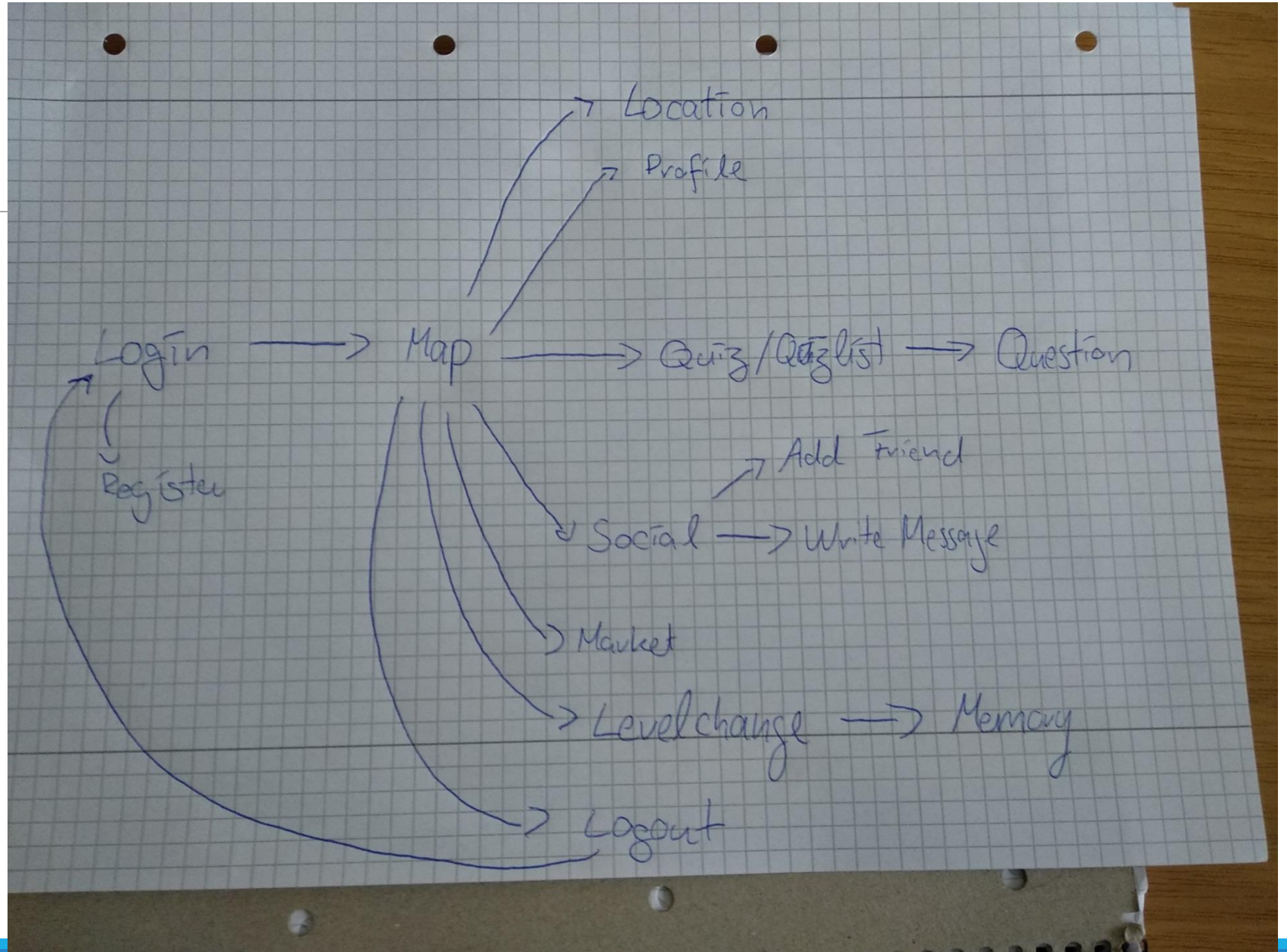
Ein Question sollte aus dem Fragetext und 4 Antwortmöglichkeiten bestehen.



2. Sitemap

SiteMap

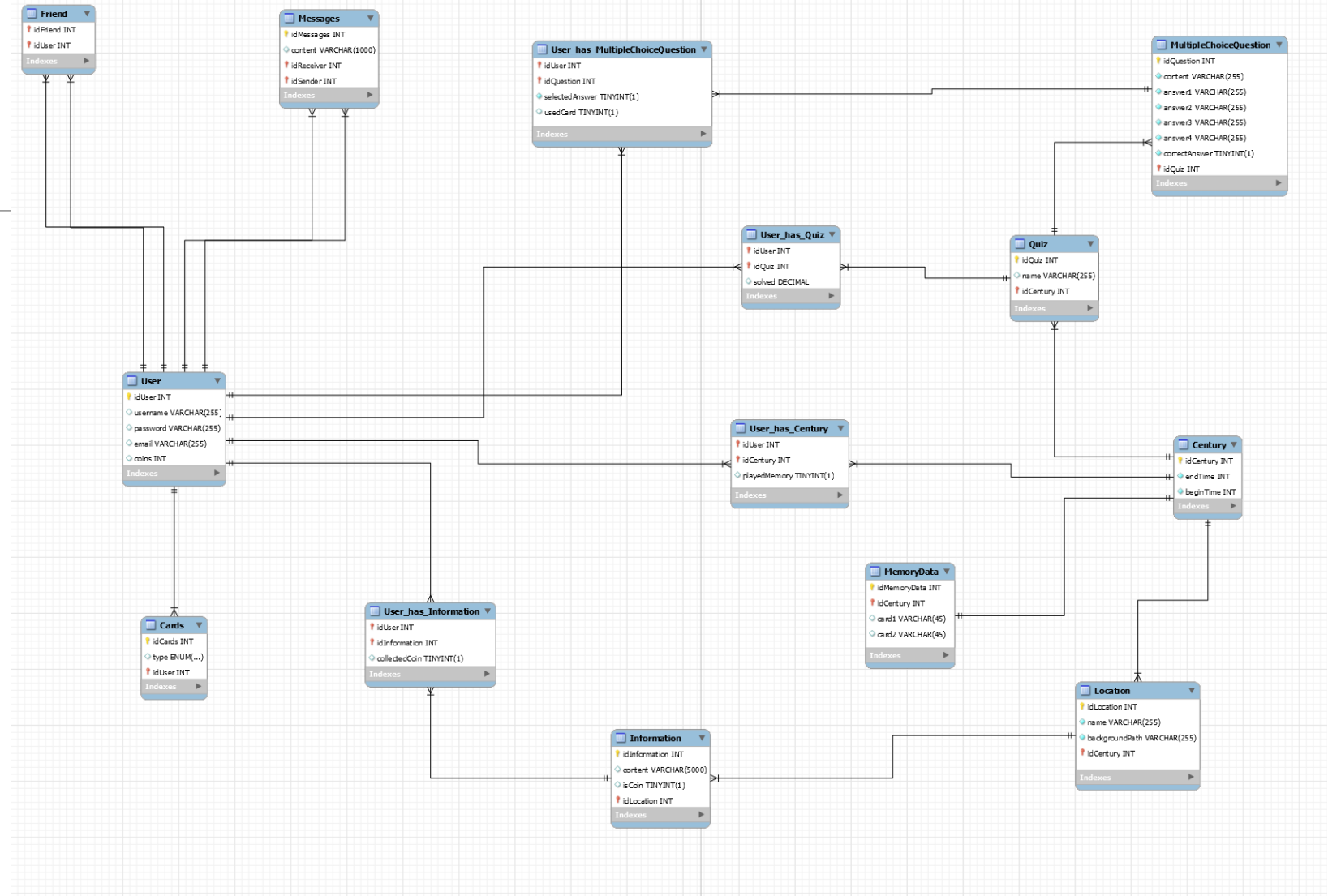
Anhand einer Sitemap sollten die Bewegungen, die ein User in der App vornehmen kann, dargestellt werden. Dies war vor allem zur Festlegung des weiteren Vorgehens und der Veranschaulichung der Hierarchie gedacht.



3. Datenbank

Datenbank

Wir erstellten dieses Datenbankschema, aus dem wir uns das SQL Script für unser Projekt generieren ließen.



4. Design und Umsetzung

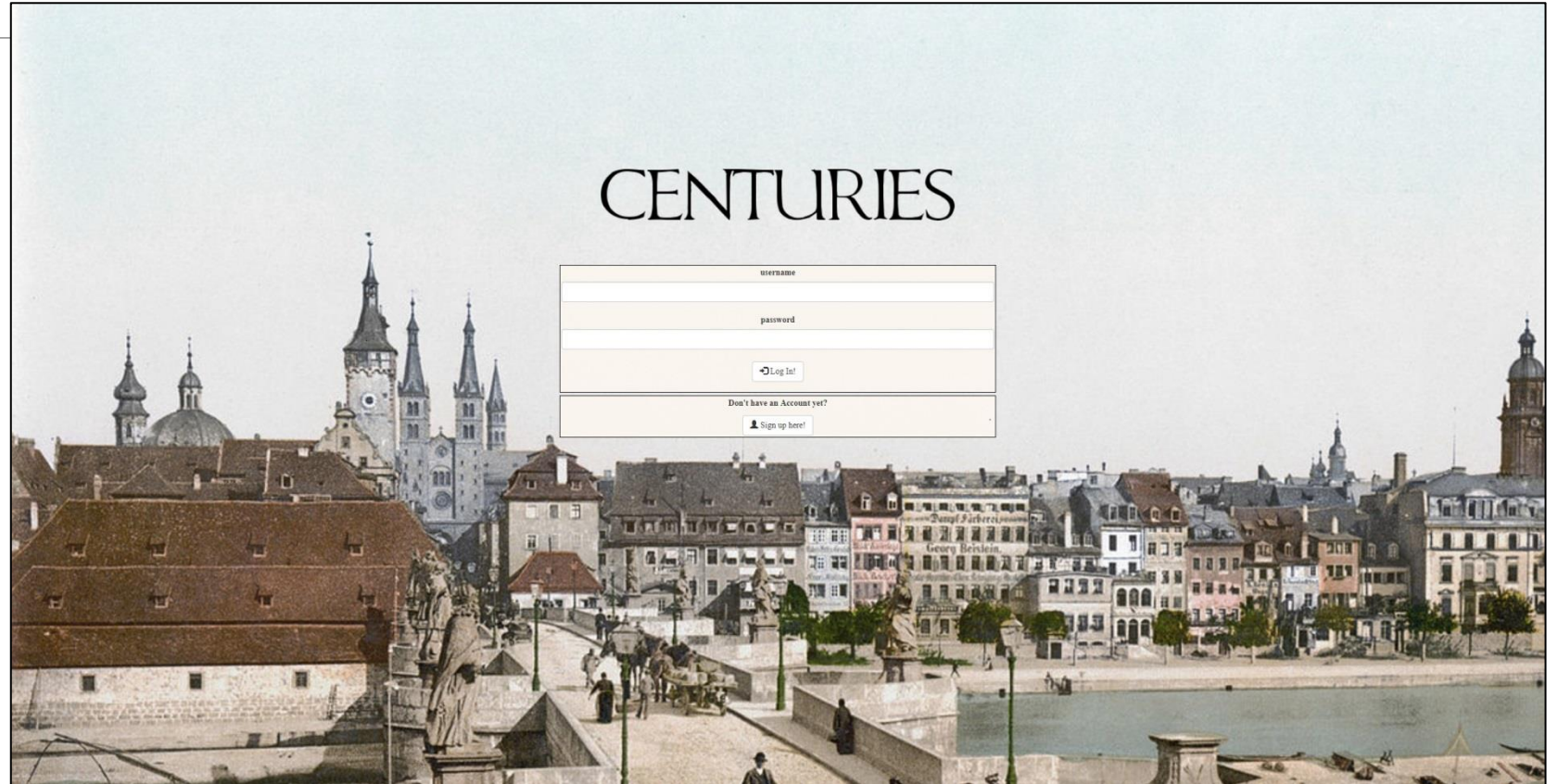
Login

Ziel des Loginbildschirms war es den Fokus auf das Login Formular zu lenken.

Außerdem sollte ein kurzer quasi-Ausblick auf das Spiel gegeben werden.

Dies wurde durch ein Hintergrundbild, das Würzburg mit einigen wichtigen Gebäuden zeigt, erreicht.

Außer dem Login sollte hier außerdem die Möglichkeit gegeben werden, auf die Registrierungsseite zu gelangen.



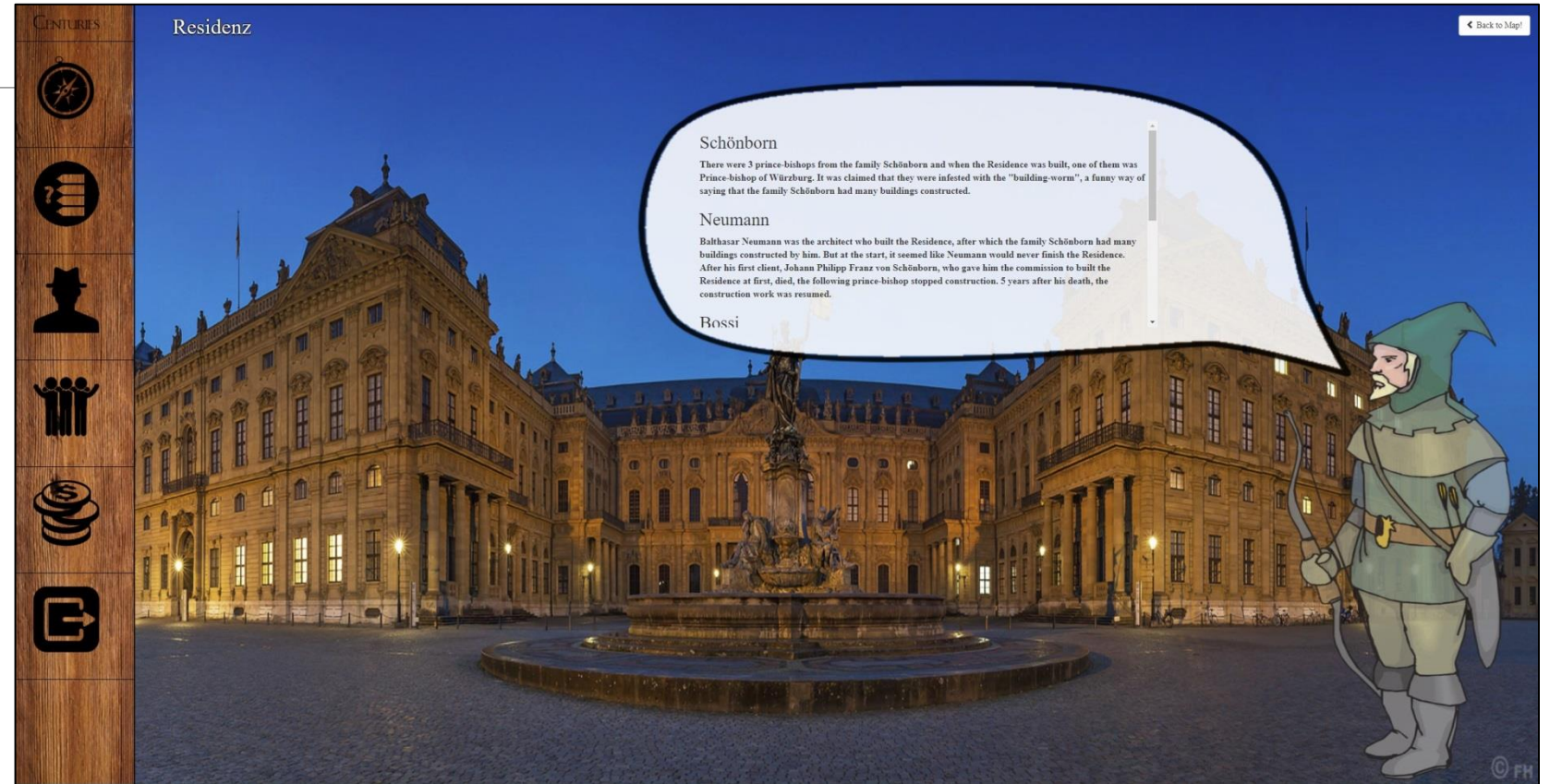
Map

Auf der Map sollten eindeutig die Locations hervorgehoben werden. Dies wurde durch die Holzschilder erreicht, die als Wegpunkte für Locations fungieren. Diese Schilder sind klickbar und leiten auf die korrespondierende Location weiter.



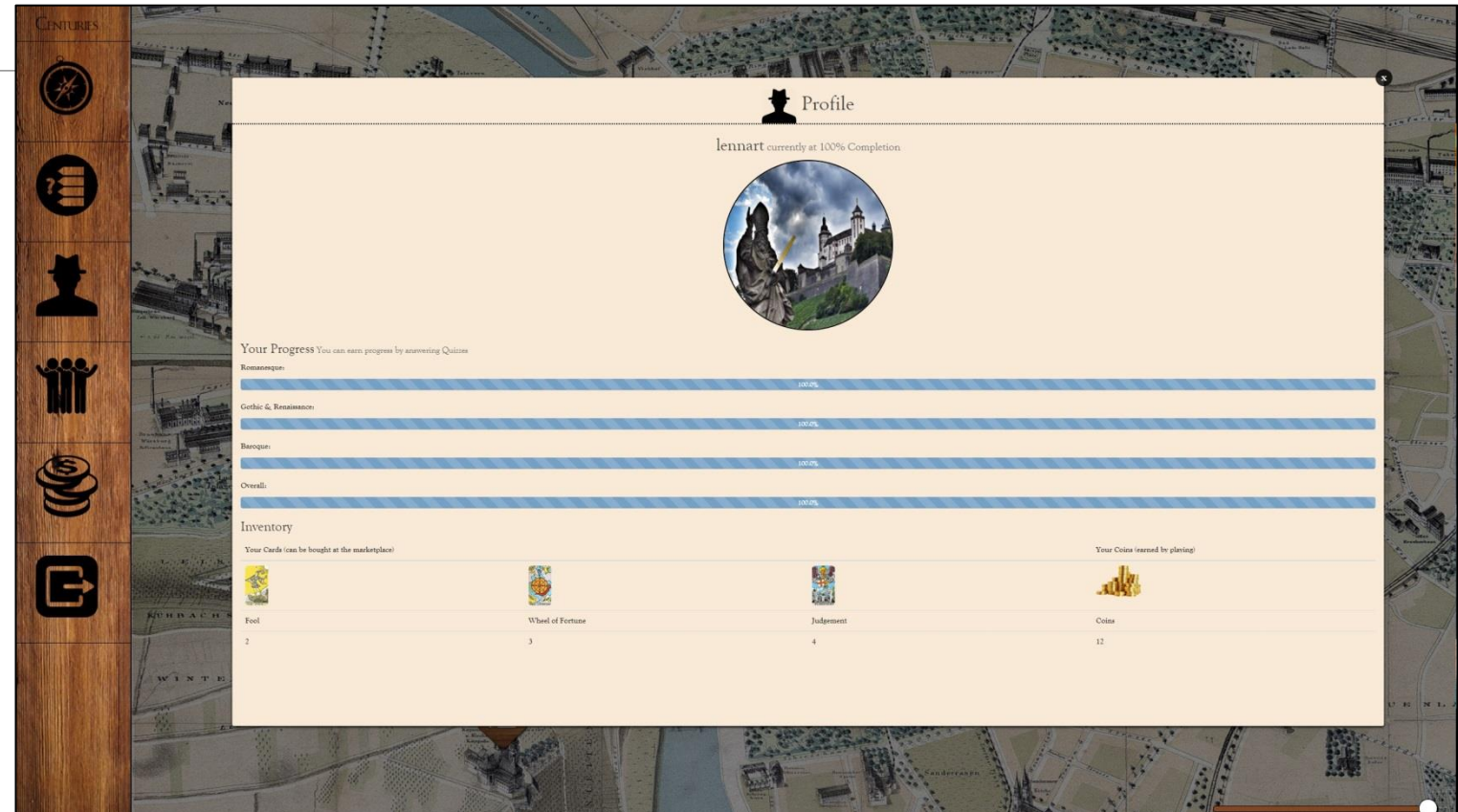
Location

Bei einer Location werden dem Benutzer vom Gästeführer Informationen zu der Location oder damit assoziierten Personen gegeben. Außerdem findet der Benutzer an bestimmten Locations Coins, mit denen auf dem Marktplatz Karten gekauft werden können.



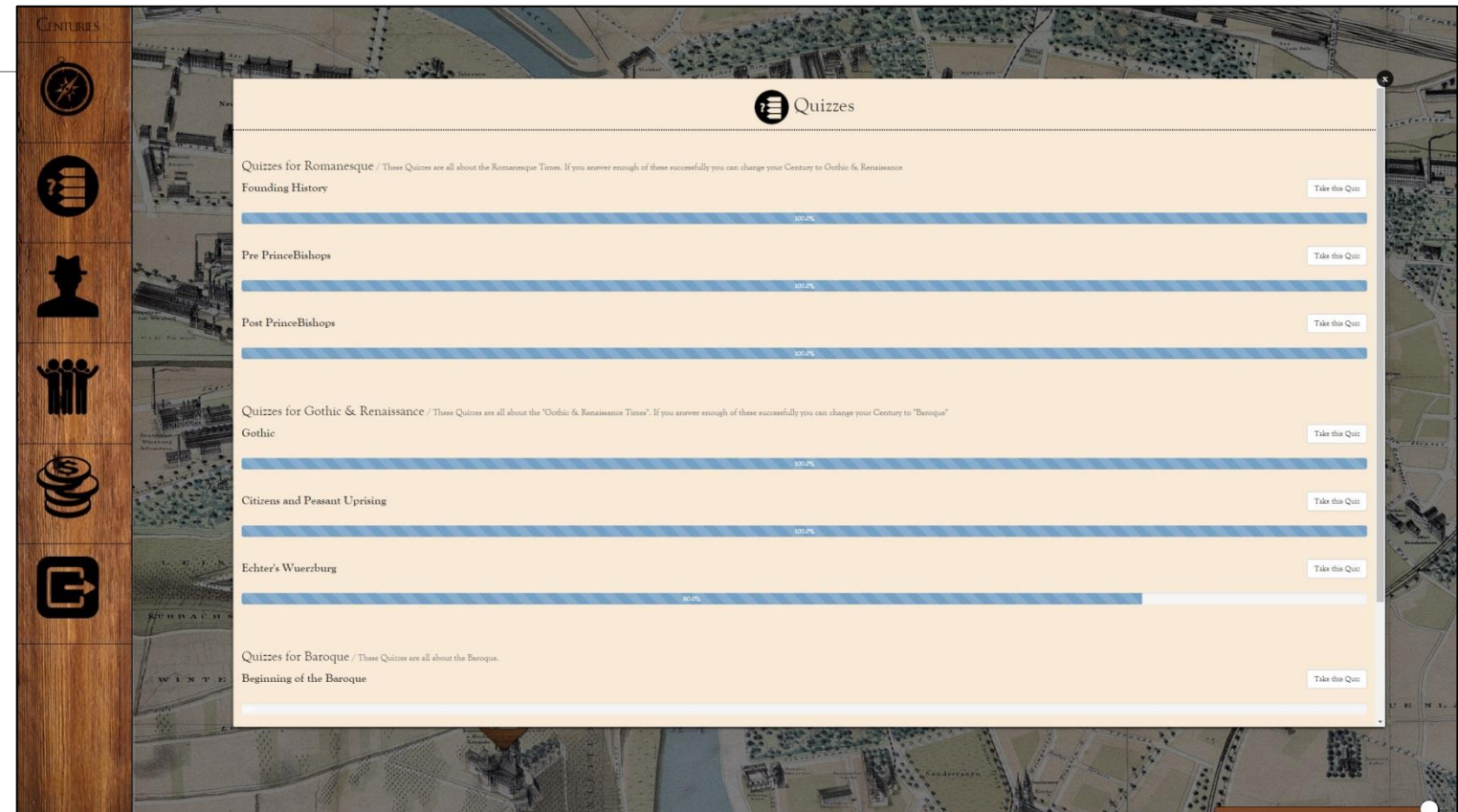
Profil

Jeder Benutzer hat ein Profil.
Auf dem Profil sieht der
Benutzer seinen
Benutzernamen zusammen mit
dem Level in dem er sich zurzeit
befindet. Außerdem kann jeder
Benutzer seinen Fortschritt und
sein Inventar sehen.



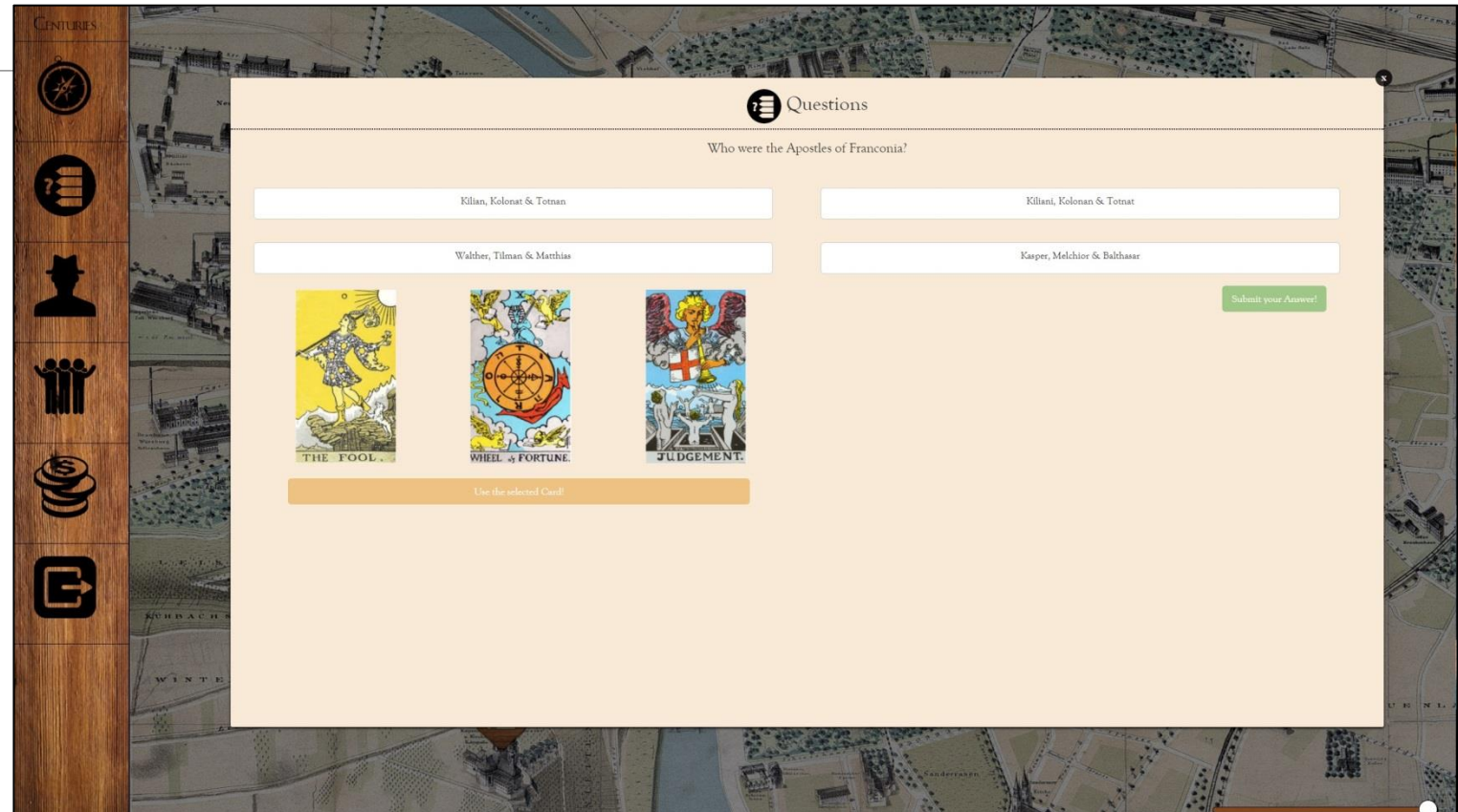
Quiz

Auf der Quizzes-Seite kann ein User seinen Fortschritt sehen und per Klick auf den Button „Take this Quiz“ das jeweilige Quiz starten.



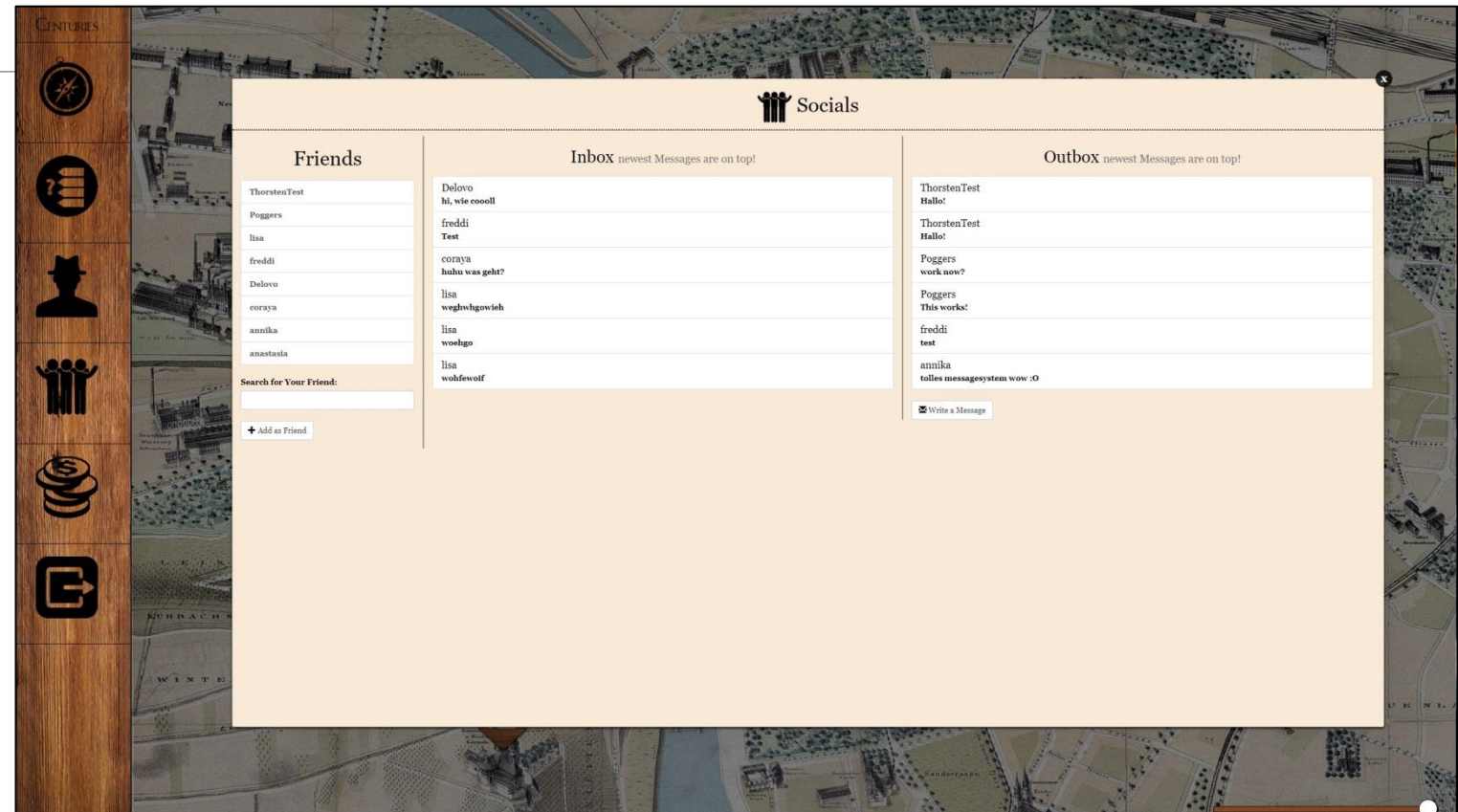
Question

Quizzes bestehen aus einigen Fragen. Für jede Frage sieht der Benutzer den Text sowie vier Antwortmöglichkeiten, von denen immer genau eine richtig ist. Der Benutzer kann seine Auswahl mit dem „Submit your Answer“ Button bestätigen. Sollte die Frage dem Benutzer zu schwer sein, kann er Karten aus seinem Inventar einsetzen, die die Beantwortung einfacher machen!



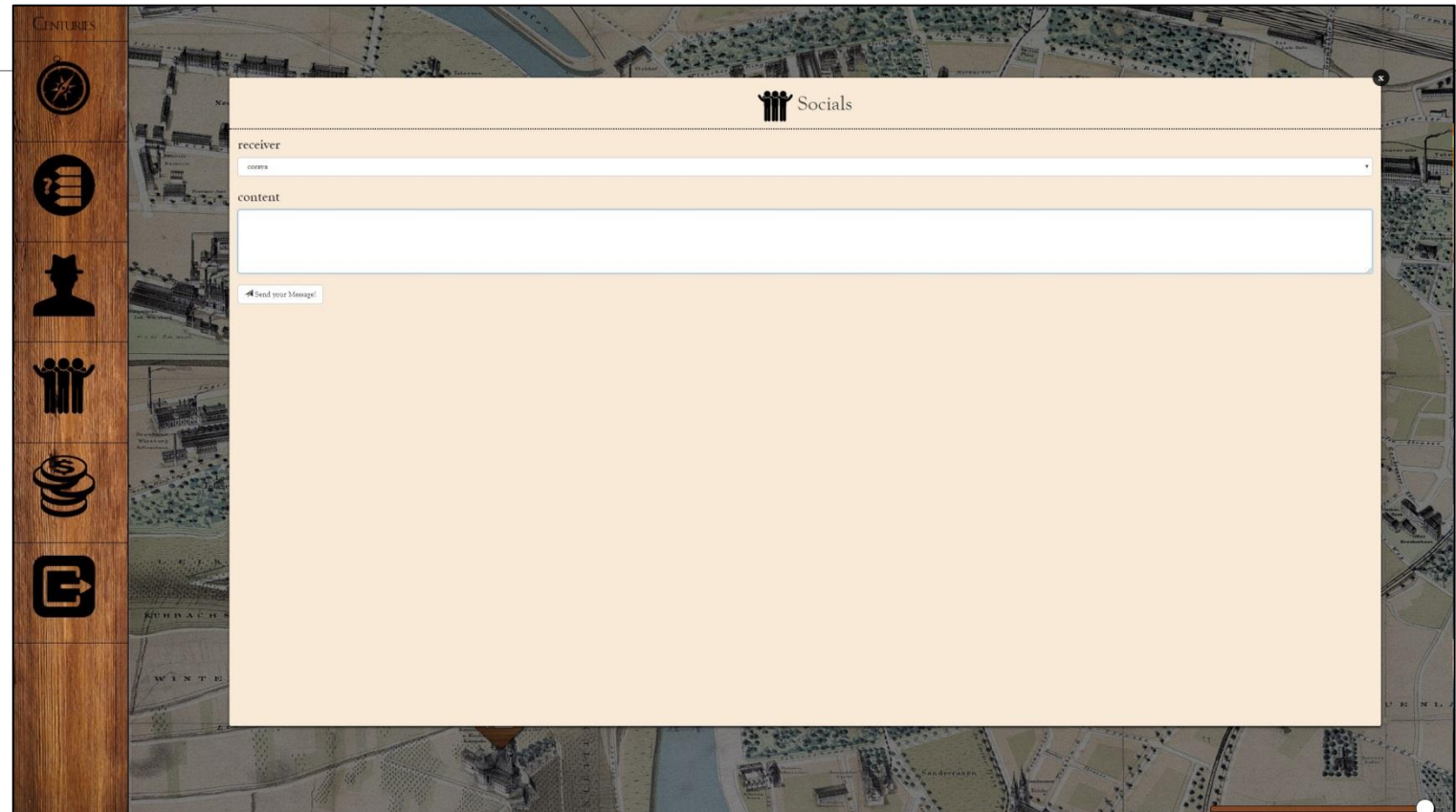
Socials

Auf der Socials-Seite kann ein Benutzer neue Freunde hinzufügen, sich per Klick auf den Namen des Freundes dessen Profil anschauen, die Nachrichten seiner Freunde sehen, sowie seinen Freunden eine Nachricht schreiben!



Message

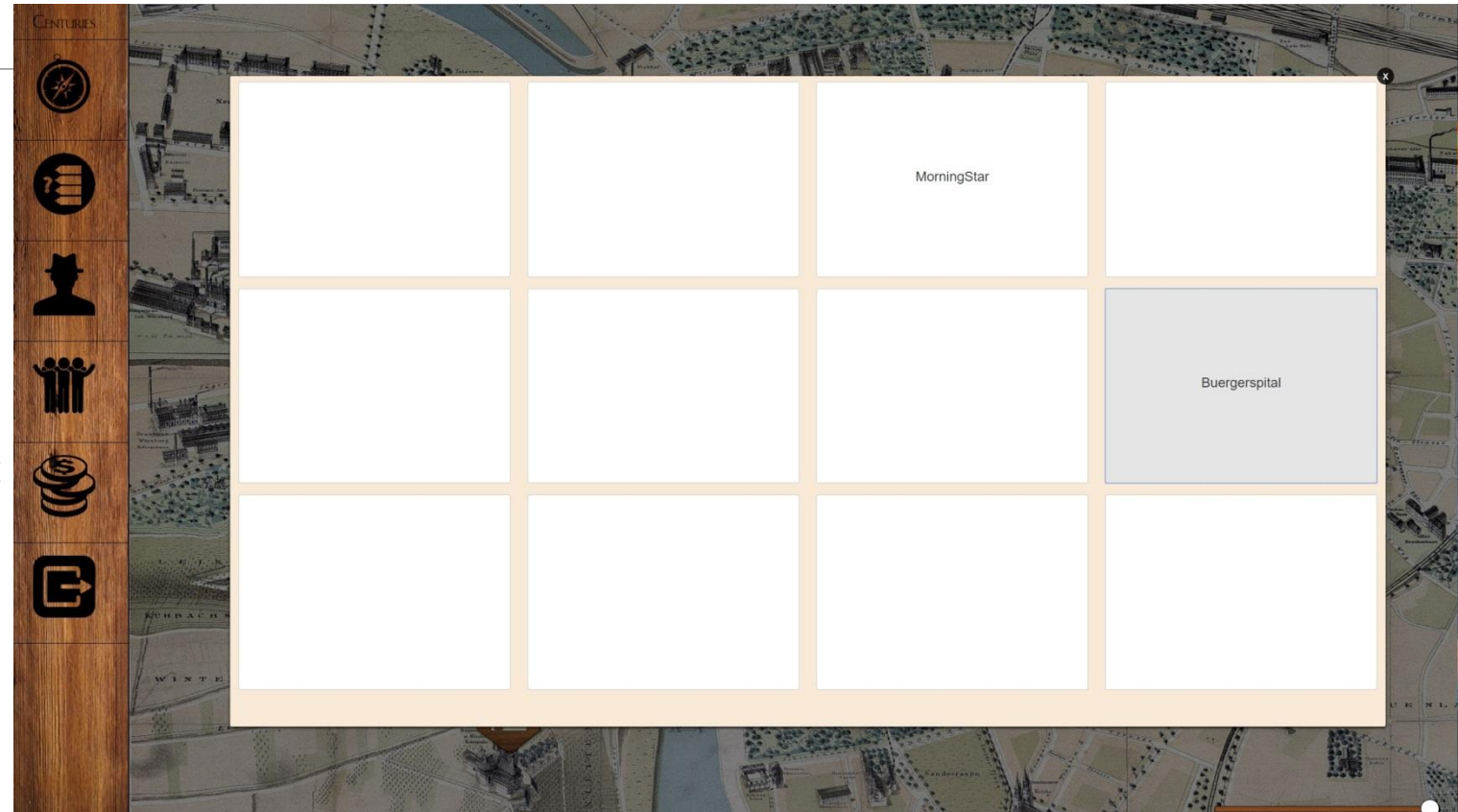
Ein Benutzer kann einen Empfänger auswählen und eine Nachricht schreiben!



Memory

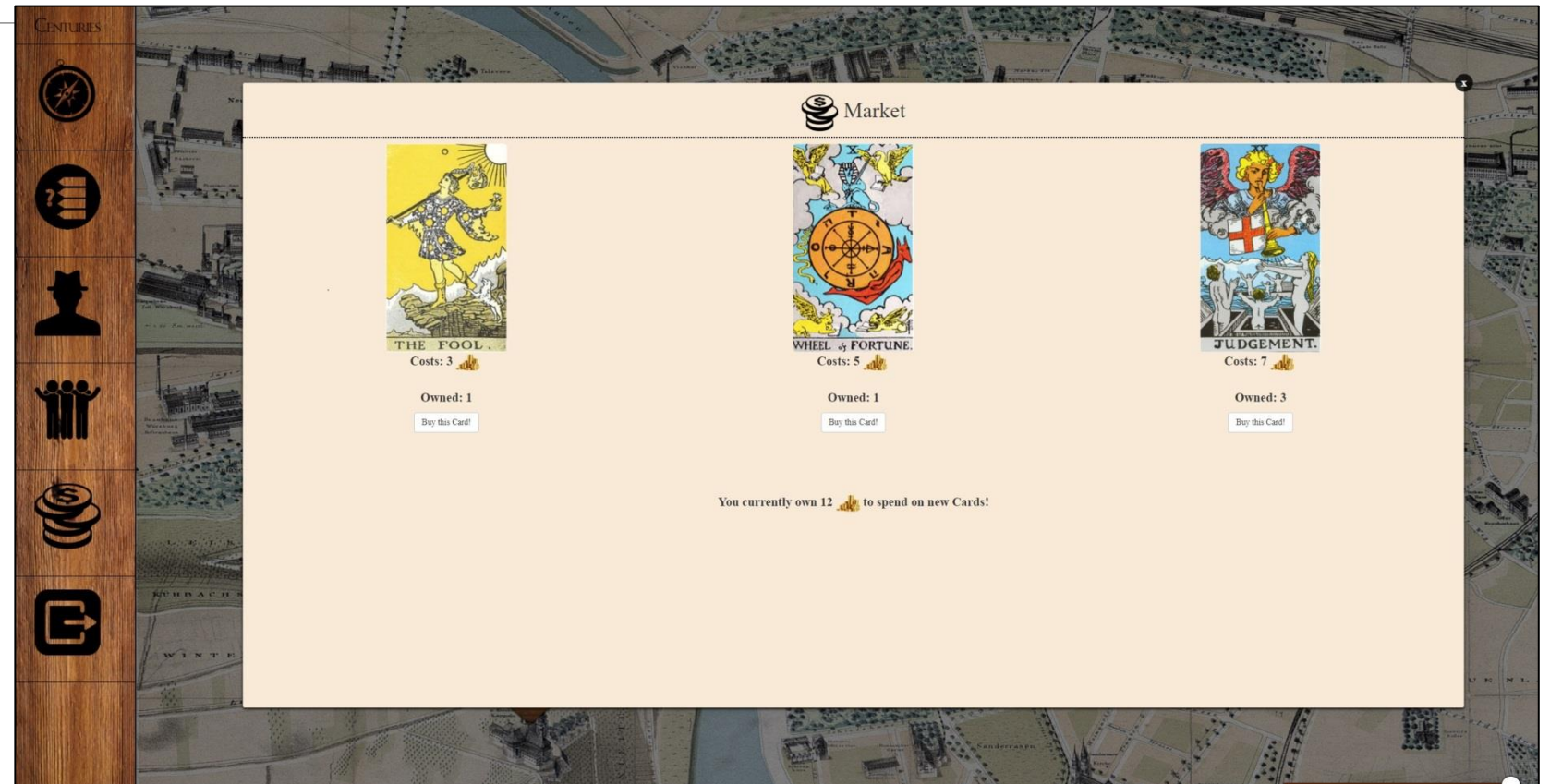
Mit dem Schieberegler unten rechts kann ein Benutzer das Level, auch als Century bezeichnet, wechseln. Dies ist jedoch nur möglich, wenn sich der Benutzer im aktuellen Century genug Wissen angeeignet hat und dies durch die Beantwortung von Quizzes nachweisen konnte.

Beim Wechseln des Levels kann der Benutzer ein Memory spielen und bekommt beim erfolgreichen Lösen Münzen gutgeschrieben.



Marketplace

Auf dem Marketplace kann der Benutzer seine gesammelten Münzen gegen Karten eintauschen, die einmalig im Quiz benutzbar sind.



5. Heuristiken

Verwendete Heuristiken - Webanwendung

#1 –Sichtbarkeit des Systemstatus

- Dem Benutzer ist durch die Beschriftung der Seiten immer klar, wo er sich befindet
- Meldungen bei unzulässigen Aktionen, Erfolgen oder Ähnlichem werden gut sichtbar und verständlich ausgegeben

#2 - Übereinstimmung zwischen System und realer Welt

- Eindeutige und verständliche Begriffe werden verwendet
- Das Menü ist klar strukturiert und bei Hover über das Menü werden die Namen der Menüpunkte angezeigt
- Verschiedene Bereiche z.B. im Menü oder im Profil sind klar von einander abgegrenzt

#3 - Benutzerkontrolle und Freiheit

- Sollte ein Benutzer aus Versehen ein Menüpunkt öffnen, kann er jederzeit das geöffnete Pop-Up mit einem Klick auf das rechts oben platzierte „X“ schließen
- Sollte ein Benutzer auf die falsche Location gekommen sein, findet er dort ebenfalls oben rechts einen Button, der zurück zur Map und damit der Startseite führt
- Zu jedem Zeitpunkt kann der Benutzer über die immer verfügbare Menüleiste auf zu den anderen Seiten gelangen

#4 - Konsistenz and Standards

- Das System ist insgesamt immer nach dem gleichen Schema gleich aufgebaut
- Alle Elemente einer bestimmten Gruppe haben dasselbe Styling
- Alle Buttons sind eindeutig beschriftet und einheitlich gestaltet
- Die Navigationsbar ist immer an derselben Stelle zu finden
- Alle Popups sind mit einem „X“ Button schließbar
- Es werden immer dieselben Schriftarten benutzt

Verwendete Heuristiken - Webanwendung

#6 - Wiedererkennen vor Erinnern

- Es werden übliche und verständliche Icons benutzt
- Der Benutzer kann eindeutig sehen, welche Buttons was tun sollten
- Es werden die Standardfarben von Bootstrap verwendet um Meldungen und Buttons zu stylen, die der einfachen Wiedererkennung dienen sollen
- Elemente des Programms, z.B. Fortschrittsbar in Profil und Quizliste / Sprechblase bei Location sind eindeutig zu erkennen und leicht zu verstehen

#8 - Ästhetik und minimalistisches Design

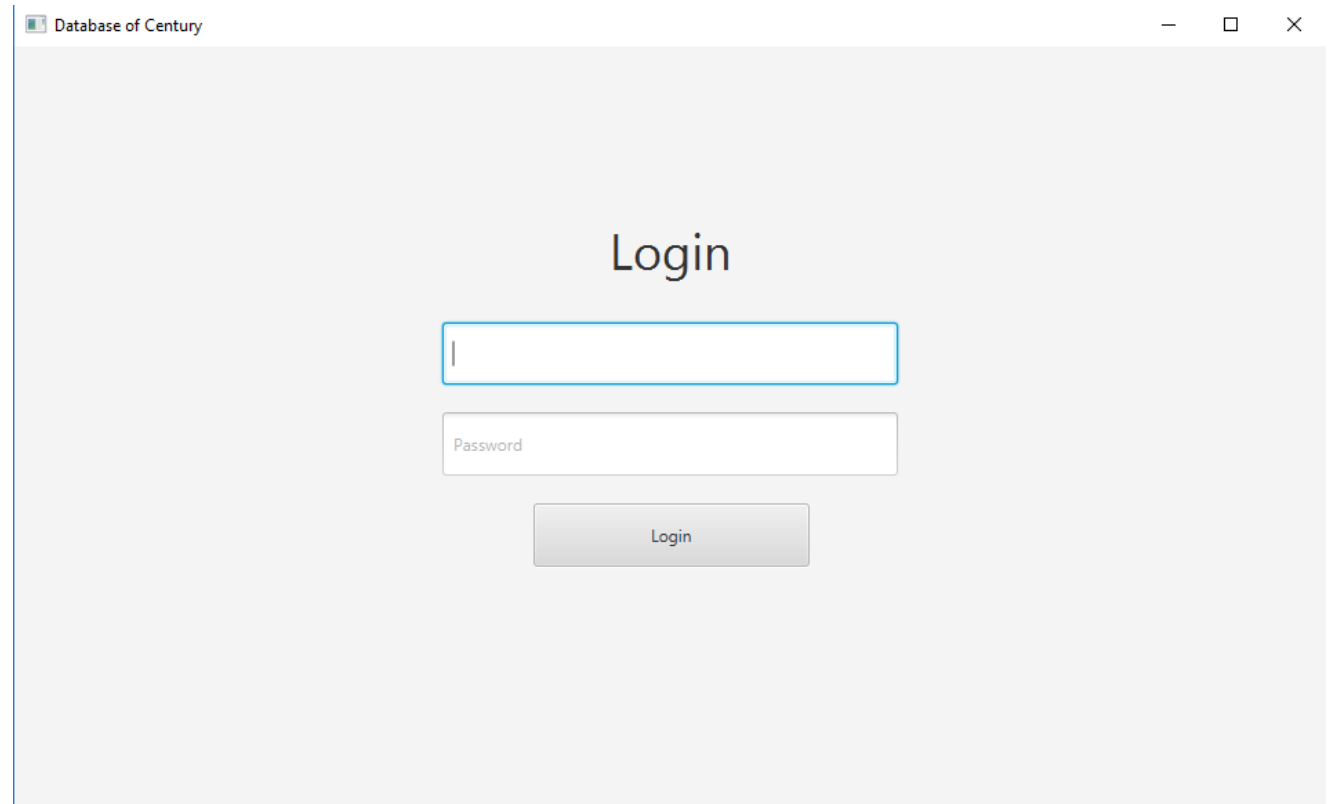
- Es wird für alle Seiten dieselbe Designgrundlage verwendet
- Minimalistische Anzahl an Elementen auf den Seiten erleichtert die Orientierung
- Die Farbgebung ist größtenteils schlicht, zentrale Elemente werden durch auffälligeres Styling hervorgehoben (z.B. Menüleiste und Wegpunkte auf der Karte)
- Es werden, außer Standardfarben von Bootstrap für Meldungen und Buttons, nur dezente Farbtöne verwendet, um die Aufmerksamkeit nicht zu sehr auf einen Punkt zu ziehen
- Wiederholung von Designelementen auf verschiedenen Seiten (z.B. Map im Hintergrund bei Profile, Quiz etc. und andauernde Orientierung am Thema Holz)

#9 - Unterstützung beim Erkennen, Verstehen und Bearbeiten von Fehlern

- Es werden bei Fehlern oder Falschbenutzung des Systems gut sichtbare und verständliche Meldungen eingeblendet
- Die Fehlermeldungen sind kurz und präzise gehalten
- Jede Fehlermeldung gibt eindeutige Informationen, was der Fehler war und schlägt vor, wie dieser zu beheben ist. Dies kann z.B. „bei dem Benutzen einer Karte ohne dass diese im Besitz des Benutzers ist“ oder „bei dem Kauf einer Karte ohne über ausreichend Münzen zu verfügen“ gesehen werden.

JavaFX Login

Der Login sollte schlicht gehalten werden und den Fokus auf das Login Form legen. Der Benutzer sollte schnell erkennen können, was er hier tun kann.



The screenshot shows a JavaFX application window titled "Database of Century". The window has a light gray background and a simple login form centered on the screen. The form consists of the word "Login" in a large, dark font, followed by two input fields: a text field for the username and a password field with a "Password" placeholder. Below the input fields is a "Login" button. The window has standard OS window controls (minimize, maximize, close) in the top right corner.

Database of Century

Login

Password

Login

JavaFX User Database

Bei der User Database sollte der Fokus auf der Tabelle und deren Bearbeitung liegen. Dies wurde erreicht indem die restliche Seite schlicht gehalten wurde. Es sollte klar ersichtlich sein, was die Buttons tun, um dem Benutzer die Bearbeitung der Datenbank zu erleichtern.

Außerdem sollte der Benutzer die Möglichkeit haben, auf die Quiz Database zu wechseln oder sich auszuloggen, wenn er mit der Bearbeitung fertig ist.

Database of Century

Database of User

change to Quiz Database Logout

Username	Password	E-mail	Coins
anastasia	anastasia	anastasia@test.de	6
annika	1234	annika@test.de	21
freddi	stoiber	freddi@test.de	2
lennart	yes	lennart@test.de	12
lisa	0987	lisa@test.de	2
Poggers	niceme.me	pog@champion.com	16
ThorstenTest	test	thorsten@test.com	6
coraya	doodlejump	lalal@web.de	50
Delovo	1234	delovo@sopra.com	1
g2-delovo	sopra	newemail@delovo.de	3
new	user	new@new.de	0
abdcde	12345	a@b.c	983

< >

username password E-mail Insert Delete

JavaFX Quiz Database

Wie bei der User Database sollte auch hier der Fokus auf den Bearbeitungsmöglichkeiten liegen. Die Quiz Database wurde ähnlich wie die User Database schlicht gestaltet, um die Verwendung zu erleichtern.

The screenshot shows a JavaFX application window titled "Database of Century". Inside, the "Quiz and Question Database" section is active. It features a "Founding History from Century with Id: 1" dropdown menu, a "Delete Quiz" button, and a "Change to User Database" button. Below these are input fields for "Name of the New Quiz" and "Choose the Century of New Quiz" (set to 1), with an "Add Quiz" button. A table displays quiz questions and answers. The table has columns: Question, Answer 1, Answer 2, Answer 3, Answer 4, and correctAnswer. The data rows are as follows:

Question	Answer 1	Answer 2	Answer 3	Answer 4	correctAnswer
Who were the Apostles of ...	Kilian, Kolonat & Totnan	Kiliani, Kolonan & Totn	Walther, Tilman & Matthias	Kasper, Melchior & Balthasar	1
What was the Apostles' of ...	The Duke didn't appro...	The Duke's wife, Gail...	They were attacked and kil...	They all succumbed to the plague	2
Who was Wuerzburg's firs...	Bonifatius	Kilian	Gosbert	Burkard	4
When did Wuerzburg bec...	742	690	1337	808	1
What made Wuerzburg a ...	The fresh air	Wuerzburg had the ...	Wuerzburg had the most s...	The bishop was said to be a gre...	3

Below the table is a horizontal scrollbar. At the bottom, there are input fields for "Question", "Answer 1", "Answer 2", "Answer 3", and "Answer 4", followed by a "Choose the right answer:" dropdown (set to 1) and "Insert" and "Delete" buttons.

Verwendete Heuristiken - JavaFX

#2 - Übereinstimmung zwischen System und realer Welt

- Aufgebaut wie eine Tabelle, in der man Werte verändern kann -> Personen die schonmal mit Excel gearbeitet haben werden das Interface sofort verstehen können
- Bezeichnungen sind eindeutig

#3 - Benutzerkontrolle und Freiheit

- Es gibt immer eine Möglichkeit aus Versehen eingesetzte Werte wieder zu löschen
- Es gibt gut erkennbare Buttons zum Wechseln der Database und für den Logout

#4 – Konsistenz und Standards

- Interface ist gleich gestaltet
- Buttons sind immer an derselben Stelle zu finden
- Zusammengehörige Elemente haben identisches Styling
- Die Farbgebung ist im ganzen Programm schlicht gehalten

Verwendete Heuristiken - JavaFX

#6 – Wiedererkennen vor Erinnern

- Bei jedem Element ist die Funktion eindeutig und sofort festzustellen
- Buttons sind als solche zu erkennen, Eingabefelder sind ebenfalls eindeutig erkennbar

#8 - Ästhetik und minimalistisches Design

- Schlichtes Design, Beschränkung auf die relevante Elemente
- Die wichtige Funktionalität ist eindeutig hervorgehoben
- Keine ablenkenden oder schwer sichtbare Elemente

Nielsen Heuristiken - Anhang

1-Sichtbarkeit des Systemstatus

Das System sollte den Benutzer immer auf dem Laufenden halten, indem es angemessenes Feedback in einer angemessenen Zeit liefert.

2-Übereinstimmung zwischen dem System und der realen Welt

Das System sollte die Sprache der Benutzer sprechen und systemorientierte Terminologien vermeiden.

Wenn man die Anweisung liest die sich im orangefarbenen Quadrat befindet, weiß man zunächst nicht was damit gemeint ist. Dieses Beispiel zeigt: wer eine Webseite betreibt, der sollte genau darauf achten wie er bestimmte Links oder Funktionen benennt. Denn wenn ein Nutzer die Bedeutung eines Hinweises nicht versteht, dann ist die Information vollkommen sinnlos.

3-Benutzerkontrolle und -freiheit ---

Ein System sollte Benutzer nie in Situationen geraten lassen, aus denen sie nicht wieder zurückfinden.

4-Konsistenz und Standards -

Benutzer sollten sich nicht über unterschiedliche

Wortwahl für gleiche Situationen oder Aktionen

wundern müssen

Der Benutzer sollte nicht über unterschiedliche Terminologien, Situationen oder Aktionen, welche eigentlich dieselbe

Sache beschreiben, stolpern.

Das o.g. Beispiel zeigt, dass Eine Anwendung den Plattformkonventionen folgt und gleiche Sachverhalte einheitlich

darstellt.

5-Fehlerverhütung

Besser als gute Fehlermeldungen ist ein gutes Design, welches das Eintreten von Fehlern erst gar nicht zulässt.

6-Wiedererkennen statt sich erinnern

Das Kurzzeitgedächtnis eines Benutzers ist begrenzt. Deshalb sollten sie sich nicht an Informationen erinnern müssen, die

in einem ganz anderen Bereich des Dialogs von Bedeutung waren.

Instruktionen für den Systemgebrauch müssen intuitiv auffindbar sein.

Dieses Beispiel zeigt, dass Software Programme den Nutzer immer intuitiv leiten müssen, durch Tooltips oder

Mouseover Effekte.

7-Flexibilität und Effizienz der Benutzung

Häufig auftretende Aktionen sollten vom Benutzer angepasst werden können, um Fortgeschrittenen eine schnellere Bedienung zu erlauben.

8-Ästhetik und minimalistisches Design

Jegliche Information sollte in einer natürlichen und logischen Ordnung erscheinen.

9-Hilfe beim Erkennen, Diagnostizieren und Beheben von Fehlern

Gute Fehlermeldungen sind defensiv, präzise und konstruktiv. Präzise Fehlermeldungen geben dem User genaue Informationen über die Ursache des Problems.

Ein Betreiber einer Webseite muss immer darauf achten, dass Fehlermeldungen nicht nur leicht verständlich sind sondern auch sofort sichtbar sind also im Fokus der Aufmerksamkeit des Nutzer liegen.

10-Hilfe und Dokumentation

Jede kleine Website sollte ohne Hilfe auskommen.