



# A Inhaltsverzeichnis

<b>A</b>	Inhaltsverzeichnis.....	II
<b>B</b>	Abbildungsverzeichnis.....	III
<b>1</b>	Einleitung & Zielsetzung .....	1
1.1	Vorgeschichte & Motivation .....	1
1.2	Zielsetzung.....	1
1.3	Aufbau der Arbeit .....	2
<b>2</b>	Idee & Konzept der App .....	3
2.1	Entstehung der Idee .....	3
2.2	Features & Funktionen.....	4
2.2.1	Das Quiz.....	4
2.2.2	Das Wörterbuch.....	7
2.2.3	Begriff des Tages .....	8
2.2.4	Die Statistik .....	9
2.2.5	Die Hilfe .....	9
2.2.6	Ändern des Wörterbuches.....	9
2.3	Layout & Navigation.....	10
2.3.1	Status Bar .....	11
2.3.2	Navigation Bar .....	12
2.3.3	View .....	13
2.3.4	Tab Bar .....	13
2.4	Didaktisches Konzept.....	14
2.4.1	Behaviorismus .....	0
2.4.2	Kognitivismus .....	0
<b>3</b>	Gestaltung der App .....	0
<b>4</b>	Programmierung der App.....	0

## B Abbildungsverzeichnis

Abbildung 1	Wireframe des Quiz-Typs Begriff zu Bilder .....	5
Abbildung 2	Wireframe des Quiz-Typs Bild zu Begriffen .....	5
Abbildung 3	Wireframe des Quiz-Typs Begriff zu Buchstaben.....	6
Abbildung 4	Wireframe des Quiz-Typs Begriff zu Eingabe.....	6
Abbildung 5	Wireframe der Wörterbuch-Übersicht.....	7
Abbildung 6	Wireframe der Wörterbuch-Details .....	7
Abbildung 7	Wireframe des Begriff des Tages .....	8
Abbildung 8	Wireframe der Statistik des Nutzers .....	8
Abbildung 9	Wireframe des App-Layouts .....	11

# 1 Einleitung & Zielsetzung

Im nachfolgenden werden die Motivation, Zielsetzung und der Aufbau der vorliegenden Bachelorarbeit erläutert.

## 1.1 Vorgeschichte & Motivation

Im Zuge des Seminars *Mobile Learning* im Wintersemester 2015/16 an der Universität Bielefeld unter Leitung von Paul John war ich, zusammen mit sieben weiteren Studierenden, an der Konzeption, Gestaltung und Umsetzung einer Lern-App für Dialektwörter aus Ostwestfalen-Lippe beteiligt. Dabei entstand ein erster funktionsfähiger Prototyp der App *OWLisch*, an dessen Programmierung ich maßgeblich mitgewirkt habe.

Bereits beim Entwickeln dieser ersten rudimentären Version habe ich einen tieferen Einblick sowohl in die Programmierung von Web-Apps mit der Programmiersprache JavaScript, als auch in die didaktische Konzeption einer solchen Lern-App erhalten. Mit dieser Arbeit wollte ich noch einen Schritt weiter gehen und aus der bereits bestehenden Idee ein fertiges Produkt erstellen, das auf aktuellen Technologien basiert und mein Verständnis für die damit verbundenen Programmiersprachen und Konzepte vertieft.

## 1.2 Zielsetzung

Das Ziel dieser Arbeit war es, eine lauffähige iOS-Version von *OWLisch* zu entwickeln. Dabei sollte die App von Grund auf neu programmiert werden; ich entschied mich allerdings dazu, die App nicht in der von Apple kreierten Programmiersprache *Swift* zu entwickeln, sondern einen Hybrid-Ansatz zu wählen, um größtmögliche Flexibilität zu erreichen.

Dabei wollte ich auf etablierte Web-Technologien wie JavaScript, HTML und CSS zurückgreifen und diese mit dem Framework *Apache Cordova* kombinieren. Dies erlaubt es, eine so erstellte App nicht nur in jedem modernen Web-Browser auszuführen, sondern diese auch mit geringem Aufwand zu einer nativen iOS- oder sogar Android-App zu konvertieren, ohne den Programmcode für jede Plattform neu schreiben zu müssen.

Der eigentliche Inhalt und das Konzept des bestehenden Prototyps sollten dabei nur geringfügig verändert oder erweitert werden; die in dieser Arbeit enthaltene Beschreibung der Konzeption und Gestaltung stimmen daher größtenteils mit der Vorgängerversion überein.

## 1.3 Aufbau der Arbeit

Diese Arbeit beschäftigt sich zunächst mit der grundlegenden Idee, der Funktionsweise, dem Aufbau und dem zugrunde liegenden didaktischen Konzept der App *OWLisch*. Nachfolgend wird die grafische Gestaltung der App näher beleuchtet und im Detail beschrieben, wobei ein Vergleich zu den *iOS Human Interface Guidelines* gezogen und eine erste Verbindung zum dazugehörigen Quellcode hergestellt wird.

Im nächsten Kapitel des Hauptteils wird die eigentliche Programmierung der App näher beschrieben; dabei werden die eingesetzten Technologien und der modulare Aufbau des Programms präsentiert. Das letzte Kapitel fasst die Arbeit zusammen und gibt einen Ausblick auf eine mögliche Fortsetzung des Projekts.

## 2 Idee & Konzept der App

*Nachfolgend wird die Idee der App, die Funktionsweise und das dahinter liegende didaktische Konzept erläutert.*

### 2.1 Entstehung der Idee

Die Idee für die App *OWLisch* entstand im Zuge des Seminars *Mobile Learning* unter Beteiligung folgender Studierenden: Tugba Aksakal, Miriam Belke, Melanie Derksen, Franziska Kluge, Lisa Kottmann, Kai-Frederik Lüking, Jakob Metzger und Philipp Niewöhner. Ziel war es zunächst, ein Konzept für eine Lern-App für Dialektwörter aus Ostwestfalen-Lippe unter Berücksichtigung von didaktischen Erkenntnissen zu entwickeln; zu diesem Zweck wurden während des Seminars interdisziplinäre Gruppen aus verschiedenen Studiengängen gebildet (interdisziplinäre Medienwissenschaften, Erziehungswissenschaften und Medieninformatik & Gestaltung).

Als Inspiration für die App dienten in erster Linie die bekannte Spiele-App *Quizduell* und das althergebrachte Karteikartensystem zum Lernen von Vokabeln. Bei mehreren Gruppendiskussionen wurden diese Beiden Konzepte vereint und konkretisiert.

Um das Lernen der ostwestfälischen Vokabeln aufzulockern, wurde ein sogenannter Gamification-Ansatz gewählt – das Anwenden spieltypischer Elemente in einem sonst spielfremden Kontext – in Anlehnung an die App *Quizduell*. Die in der App enthaltenen Begriffe sollten als ein Quiz präsentiert werden, das über mehrere Schwierigkeitsstufen verfügt und den Spieler bei richtigen Antworten mit einem kleinen Erfolg belohnt.

Im Hintergrund, für den Spieler weitestgehend unsichtbar, sollte ein einfaches Karteikartensystem implementiert werden; Begriffe, die falsch erraten werden, sollten häufiger im Quiz auftauchen und so in den Vordergrund rücken, bis der Spieler die entsprechende Frage richtig beantwortet.

## 2.2 Features & Funktionen

Nachdem die Kernidee der App ausformuliert wurde, konnten konkrete Funktionalitäten erdacht und skizziert werden. Im Fokus stand dabei zunächst das Quiz als Hauptbestandteil der App; dieses sollte über mehrere Quiz-Typen verfügen, die verschiedene Schwierigkeitsgrade darstellen und dem Spieler sowohl Abwechslung, als auch eine Herausforderung bieten.

### 2.2.1 Das Quiz

In der ersten und leichtesten Stufe wird dem Spieler zunächst eine simple Multiple-Choice-Frage über einen Begriff mit vier Antwortmöglichkeiten präsentiert. Dies ermöglicht es dem Spieler, durch einfaches Raten auf die richtige Lösung zu kommen. Dieser Typ teilt sich wiederum in vier Sub-Typen auf:

- Ein ostwestfälischer Begriff mit vier hochdeutschen Begriffen als Antwortmöglichkeit (z.B. „Was heißt Pinneken?“).
- Ein hochdeutscher Begriff mit vier ostwestfälischen Begriffen als Antwortmöglichkeit (z.B. „Was heißt Schnapsglas?“).
- Ein ostwestfälischer Begriff mit vier Bildern als Antwortmöglichkeiten (z.B. „Was heißt Schnapsglas?“); siehe Abbildung 1.
- Ein Bild zu einem Begriff mit vier ostwestfälischen Begriffen als Antworten; siehe Abbildung 2.

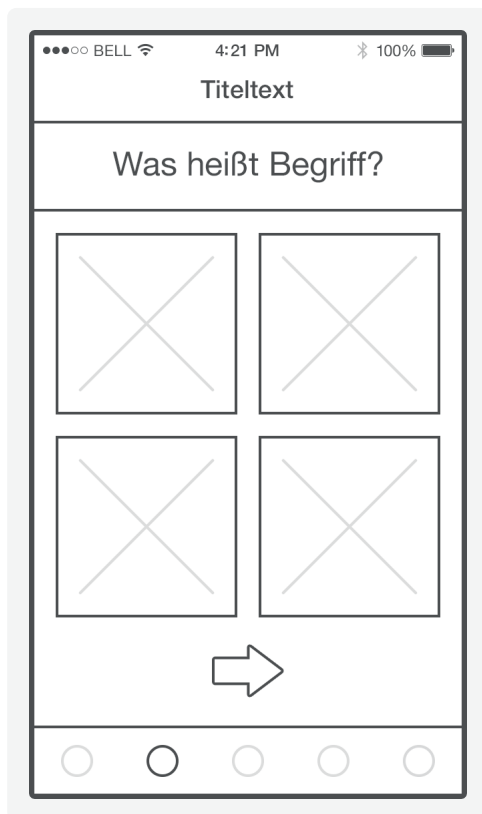


Abbildung 1

Wireframe des Quiz-Typs  
Begriff zu Bilder

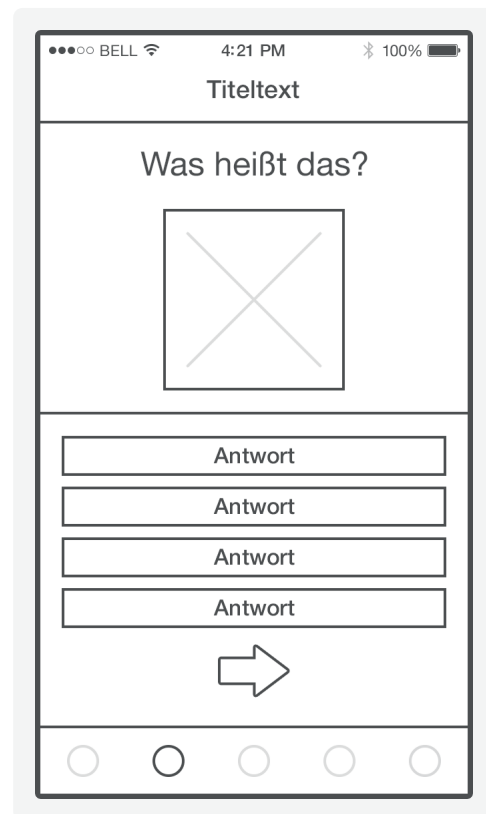


Abbildung 2

Wireframe des Quiz-Typs  
Bild zu Begriffen

Diese vier Typen sollten dabei jedes Mal zufällig ausgewählt werden. Die Bilderrätsel lockern dabei einerseits die sonst stark textbasierte App auf und unterstützen andererseits visuelles Lernen und können dem Spieler so eine zusätzliche Erinnerungsstütze bieten, wenn der entsprechende Begriff noch einmal im Quiz auftaucht.

Wird ein Begriff während dieser ersten Stufe richtig erraten, steigt der Spieler für diesen um eine Stufe auf und erhält beim nächsten Vorkommen einen neuen Schwierigkeitsgrad. Bei dieser zweiten Stufe wird dem Spieler die hochdeutsche Übersetzung des Begriffs präsentiert (z.B: „Was heißt Schnapsglas?“), während alle im ostwestfälischen Wort vorkommenden Buchstaben zufällig gemischt dargestellt sind und in die richtige Reihenfolge gebracht werden müssen.



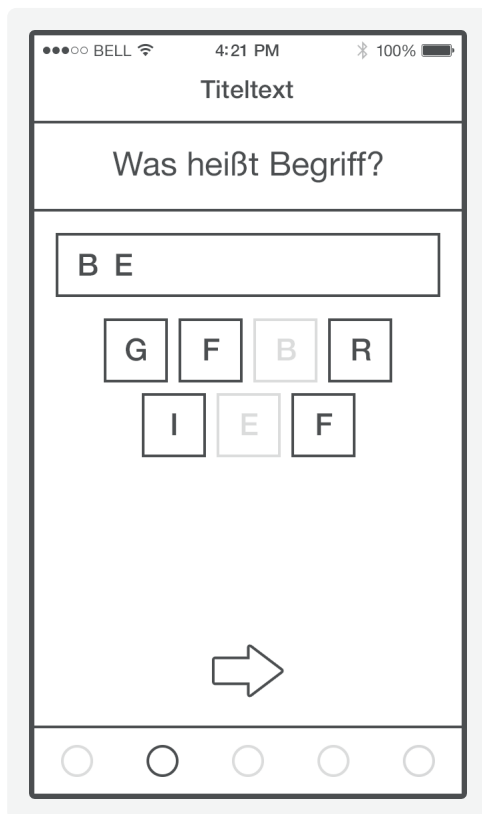


Abbildung 3

Wireframe des Quiz-Typs  
Begriff zu Buchstaben

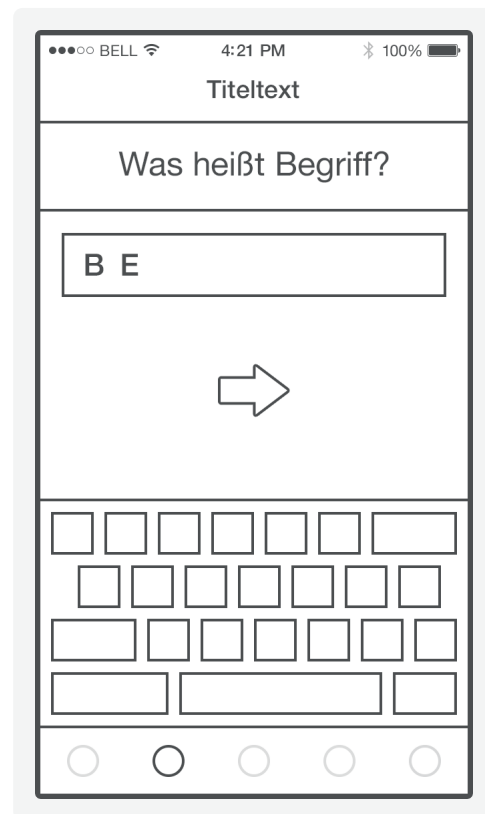


Abbildung 4

Wireframe des Quiz-Typs  
Begriff zu Eingabe

Hierdurch soll der Spieler die korrekte Schreibweise des Begriffes erlernen, erhält durch die Vorgabe der Buchstaben allerdings weiterhin eine Hilfestellung, sodass die Lösung durch Knobeln und ausprobieren erreicht werden kann (siehe *Abbildung 3*). Über einen vorhandenen „Lösen“-Button kann die Eingabe bestätigt werden.

Die letzte und somit schwierigste Stufe ähnelt ihrem Vorgänger; auch hier wird die hochdeutsche Übersetzung dargestellt, es entfällt jedoch die Hilfestellung. Der Spieler muss den ostwestfälischen Begriff selbstständig mittels einer eingeblendeten Tastatur korrekt eingeben. Kommt ein Spieler hier auf die richtige Lösung, so wurde der Begriff erfolgreich erlernt (siehe *Abbildung 4*).

## 2.2.2 Das Wörterbuch

Ein weiteres geplantes Feature für die App ist das persönliche Wörterbuch des Spielers. Hier sollen automatisch alle Begriffe hinzugefügt werden, die im Quiz richtig erraten wurden; so kann der Nutzer bereits kennengelernte Begriffe erneut nachschlagen und zusätzliche Information darüber erhalten.

Das Wörterbuch verfügt sowohl über eine Übersichtsliste aller Begriffe (siehe *Abbildung 5*), als auch über Detailansichten, die vorhandene Bilder und Herkünfte, Zitate und Übersetzungen über den Begriff enthalten (siehe *Abbildung 6*). Zusätzlich soll hier eine auditive Komponente ergänzt werden, indem alle Begriffe als eingesprochene Audiodateien im Wörterbuch abgespielt werden können, sodass der Nutzer die korrekte Aussprache der Wörter lernen kann.

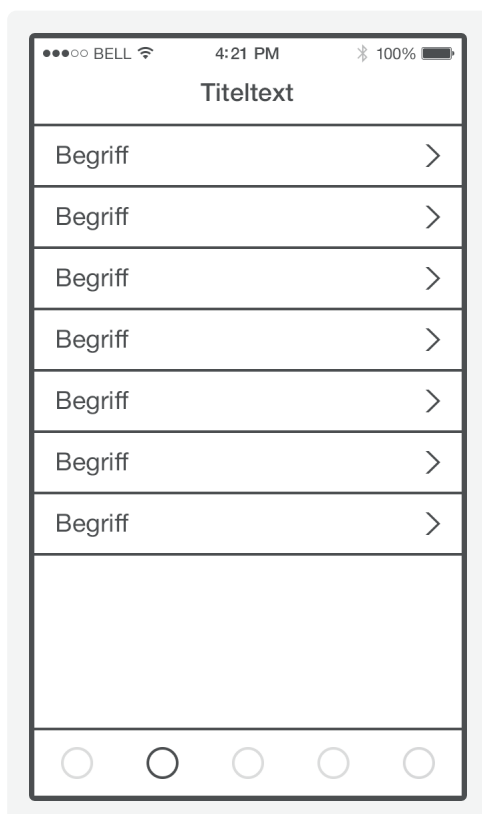


Abbildung 5

Wireframe der Wörterbuch-Übersicht



Abbildung 6

Wireframe der Wörterbuch-Details

### 2.2.3 Begriff des Tages

Um dem Nutzer beim Starten der App einen ersten Eindruck über die Inhalte zu verschaffen und nicht sofort in einen der Menüpunkte zu werfen, kam die Überlegung auf, den *Begriff des Tages* als Startbildschirm einzusetzen. Dieser Bildschirm präsentiert dem Nutzer einen der in der App vorhandenen Begriffe mitsamt Bild, Übersetzung und Informationstext, sodass man bereits vor dem ersten Spielen des Quiz ein neues Wort kennenlernen kann (siehe *Abbildung 7*).

Dieser Begriff wird täglich zufällig gewechselt und kann dem Nutzer somit sowohl bereits bekannt, als auch völlig neu sein. Gelernte Begriffe können so noch einmal aufgefrischt werden, sodass sie beim nächsten Spiel einfacher zu lösen sind.

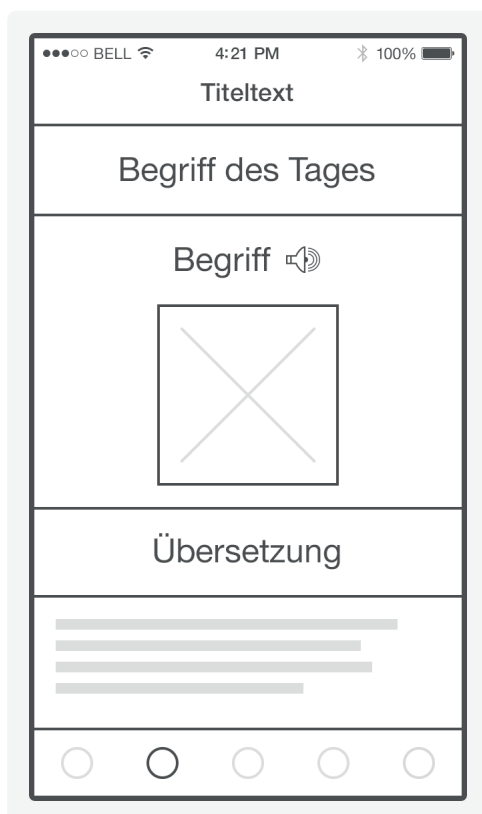


Abbildung 7

Wireframe des Begriff des Tages

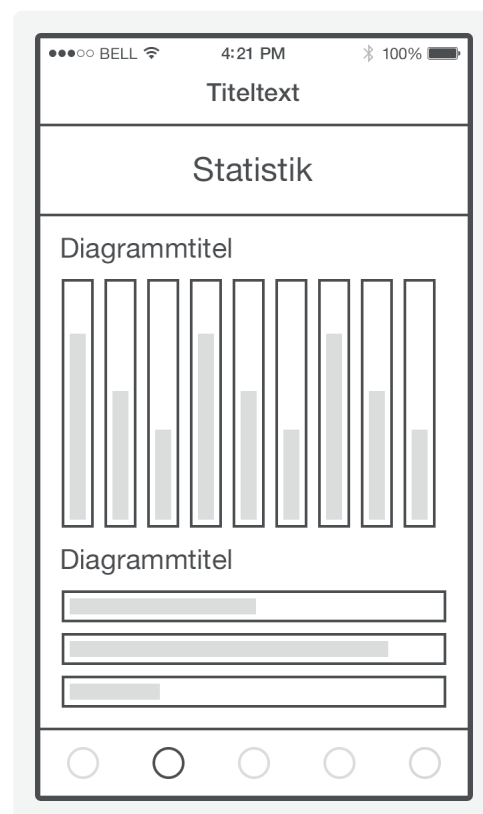


Abbildung 8

Wireframe der Statistik des Nutzers

### 2.2.4 Die Statistik

Damit der Nutzer der App zu jeder Zeit einen Überblick über den bisher erzielten Lernfortschritt behält, wurde beschlossen, eine persönliche Statistik zu implementieren, in der über Diagramme wichtige Informationen dargestellt werden (siehe Abbildung 8).

Hier erhält man Einsicht in die erzielten Punkte aus den letzten Quiz-Spielen, die Anzahl der bisher gelernten Begriffe in Relation zur Gesamtzahl der Begriffe und eine Aufschlüsselung der aktuellen Stufen der gelernten Begriffe. Auf diese Weise soll dem Nutzer der Fortschritt visuell vor Augen geführt werden, sodass ersichtlich ist, was bereits erreicht wurde und wie viel noch fehlt, bis alle Begriffe erfolgreich erlernt wurden.

### 2.2.5 Die Hilfe

Da trotz sorgfältiger Überlegungen hinsichtlich der Inhalte und Funktionsweisen der App beim Nutzer dennoch Fragen zur Bedienung aufkommen können, sollte zusätzlich noch ein ausführlicher Hilfetext in der App untergebracht werden. Hier sollen die wesentlichen Funktionen vom Quiz, dem Wörterbuch und der Statistik verständlich erklärt werden.

### 2.2.6 Ändern des Wörterbuches

Es kam bereits während der Planungsphase die Idee auf, die App nicht nur auf Ostwestfälische Begriffe zu beschränken, sondern auch andere deutsche Dialekte zu unterstützen. Dieses Feature wurde zwar bei der Entwicklung des ersten Prototypen verworfen, bei der finalen Umsetzung jedoch wieder aufgegriffen; näheres dazu findet sich in Kapitel 3 und Kapitel 4.

## 2.3 Layout & Navigation

Nachdem geklärt war, welche Grundfeatures die App enthalten soll und wie diese in etwa gestaltet sein sollten, musste das Konzept noch um eine Navigation und Inhaltsstruktur erweitert werden. Aufgrund der vorhandenen persönlichen Geräte und der bereits gemachten Erfahrungen der Gruppenmitgliedern mit unterschiedlichen Smartphones und Betriebssystemen, wurde iOS als Zielplattform für den ersten Prototypen ausgewählt, mit dem iPhone 5 (und später dem iPhone SE) als Testgerät.

Als primäre Quelle für das grundlegende Layout und die Navigationsstruktur der App wurden die *iOS Human Interface Guidelines*<sup>1</sup> herangezogen; hier sind sämtliche von Apple empfohlenen Interface-Elemente für Apps dokumentiert und beschrieben. Das Einhalten von Richtlinien und Empfehlungen ist beim Entwerfen einer App von erheblicher Bedeutung, um dem Endnutzer eine intuitive Bedienung der App zu ermöglichen und sich auf den Inhalt zu konzentrieren. Auch andere beliebte Apps wie *Facebook* und *WhatsApp* dienten der Gruppe als Inspiration für die Gestaltung des Layouts.

Für die Inhaltsstruktur wurde dabei eine flache Hierarchie gewählt, die mit minimalen Verschachtelungen und möglichst wenigen Menüs auskommt, damit der Nutzer stets die Übersicht behält, wo er sich gerade in der App befindet. Die verwendeten Layout- und Navigationselemente und ihre Beziehung zu den zuvor beschriebenen Features werden im folgenden im Detail beschrieben und bildlich dargestellt (siehe Abbildung 9), auf die finale Gestaltung dieser Elemente wird in Kapitel 3 eingegangen.

---

1. Vgl. Apple Inc.: „iOS Human Interface Guidelines“, unter:  
<https://developer.apple.com/ios/human-interface-guidelines> (abgerufen am 09.11.2016)

### 2.3.1 Status Bar

Die *Status Bar* ist in iOS-Apps ein statischer und weitestgehend unveränderlicher Bestandteil, der sich stets am oberen Rand des Bildschirms befindet und wichtige Systeminformationen wie die Netzwerk-Konnektivität, die Uhrzeit und den aktuellen Batterie-Ladestand darstellt.

Dabei ist der Inhalt dieser Leiste global gültig und damit unabhängig von der gerade geöffneten App. Die einzige Ausnahme, die die *iOS Human Interface Guidelines* erlauben, liegt bei Spiele-Apps vor, die im Vollbildschirm-Modus betrieben werden; hier kann es sinnvoll sein, die *Status Bar* vollständig auszublenden, was bei *OWLisch* allerdings nicht zutrifft. Daher muss beim Gestalten des App-Layouts Platz für diese Leiste eingeräumt werden.

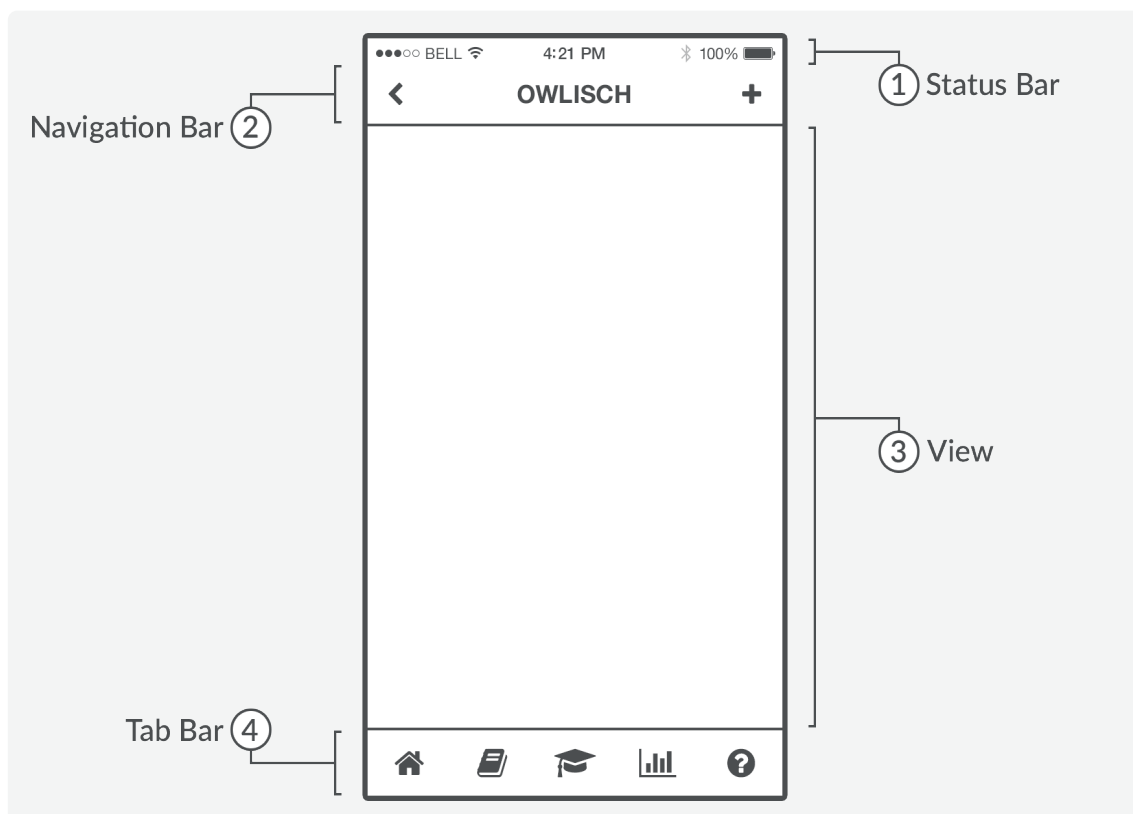


Abbildung 9

Wireframe des App-Layouts

### 2.3.2 Navigation Bar

Die *Navigation Bar* ist in iOS von wesentlicher Bedeutung für die Navigation innerhalb der App. In der Regel ist diese Leiste visuell mit der zuvor erwähnten *Status Bar* verknüpft und enthält kontextabhängige Informationen über den aktuellen Inhalt der App. Hier kann beispielsweise der Titel der gerade gewählten View (siehe Punkt 2.3.3) angezeigt werden. Auch ist es üblich und empfohlen, auf der linken Seite der Leiste einen Zurück-Button (z.B. als Pfeilsymbol) zu platzieren, um innerhalb des aktuellen Kontextes zurück zu gelangen. Dies trifft bei *OWLisch* vor allem auf das Wörterbuch zu, innerhalb dessen man sich zwischen der Begriff-Liste und den Begriff-Details vor und zurück navigieren können soll (siehe 2.2.2).

Des Weiteren lassen sich in der *Navigation Bar* auch zusätzliche Buttons unterbringen, um kontextabhängige Aktionen auszuführen; im Wörterbuch könnten hier Buttons in Form von Symbolen sein, um die Begriff-Liste zu sortieren oder mittels einer Suchfunktion zu filtern, während im Quiz ein Abbrechen-Button platziert werden kann, mit dem das bereits gestartete Quiz beendet werden kann.

Die Platzierung der *Navigation Bar* ist statisch, sie kann also nicht beliebig verschoben werden; auch sollte sie zu jeder Zeit sichtbar sein, um dem Nutzer stets die Übersicht über seine aktuelle Position innerhalb der App zu gewähren. Für Apps, die einen großen Fokus auf die bloße Darstellung des Inhaltes legen, kann es sinnvoll sein, die *Navigation Bar* gelegentlich auszublenden, damit mehr Platz zur Verfügung steht (*Facebook* und *Safari* blenden die Leiste z.B. aus, wenn der Nutzer nach unten scrollt und wieder ein, sobald wieder nach oben gescrollt wird); für *OWLisch* ist dies jedoch nicht notwendig.

### 2.3.3 View

Die *View* ist das Herzstück der App, denn hier wird der eigentliche Inhalt angezeigt. Alle in *Kapitel 2.2* skizzierten Features werden hier platziert und abhängig vom gerade gewählten Menüpunkt (siehe 2.3.4) ein- und ausgeblendet. Dieses Element nimmt daher auch den größten Platz auf dem Bildschirm ein und lässt sich scrollen, wenn der dargestellte Inhalt über den vorhandenen Platz hinausgeht. Innerhalb der *View* kann der Nutzer unter Umständen ebenfalls navigieren, z.B. durch Auswählen eines Begriffes innerhalb der Begriff-Liste im Wörterbuch. Der Inhalt der zuvor beschriebenen *Navigation Bar* hängt maßgeblich davon ab, welcher Inhalt gerade in der *View* dargestellt wird. Im Grunde ist dieses Element nichts weiter als ein leerer Kasten und ist daher selbst in keiner Weise gestaltet.

### 2.3.4 Tab Bar

Die *Tab Bar* ist ein Standardelement aus den *iOS Human Interface Guidelines*, das in einer Vielzahl von Apps verwendet wird, darunter auch die meisten Apps von *Apple* selbst. Sie stellt die primäre Navigation für die App dar und befindet sich stets am unteren Rand des Bildschirms. In dieser Leiste können bis zu fünf Icons untergebracht werden, die einerseits eine Navigation zwischen den dargestellten Menüpunkten erlauben und andererseits durch die visuelle Hervorhebung des aktuell gewählten Icons stets anzeigen, wo man sich als Nutzer gerade befindet (ergänzend zum Titel in der *Navigation Bar*).

Dies gewährleistet die eingangs erwähnte flache Hierarchie der App; die fünf vorhandenen Menüpunkte liegen alle gleichberechtigt und ohne Verschachtelung auf der selben Navigations-Ebene. In *OWLisch* erhalten die fünf wichtigsten Features der App (siehe 2.2) jeweils ein eigenes Icon in der *Tab Bar*.



Das erste Icon verweist dabei auf den *Begriff des Tages*, danach folgen das *Wörterbuch*, das *Quiz*, die *Statistik* und schließlich die *Hilfe*. Falls weitere (weniger wichtige) Menüpunkte hinzukommen sollten, lässt sich der letzte Tab in der Leiste zum Menüpunkt „Mehr“ abwandeln, der dann eine Liste aus weiteren, diesem Punkt untergeordneten Optionen (z.B. die *Hilfe* oder das noch im Raum stehende Feature zum Ändern des Wörterbuches) enthält. Diese Vorgehensweise wird auch von Apple vorgeschlagen, da so nicht nur für den Nutzer, sondern auch für den Entwickler während des Planungsprozesses klar wird, welche Funktionen der App am wichtigsten sind und daher am prominentesten dargestellt werden sollten.

Die *iOS Human Interface Guidelines* empfehlen außerdem, die *Tab Bar* stets anzuzeigen, unabhängig von der aktuellen Position innerhalb der Navigation, um dem Nutzer stets die Option zu bieten, zu jedem beliebigen Punkt zu navigieren. Gegen diese Regel wird in *OWLisch* beim Quiz allerdings bewusst verstoßen werden (*siehe Kapitel 3*); sobald ein Quiz gestartet wird, soll es dem Nutzer nämlich explizit nicht mehr erlaubt sein, einen anderen Menüpunkt auszuwählen, so lange das Quiz nicht beendet oder abgebrochen wurde. Dies soll verhindern, dass der Nutzer sich selbst betrügt, indem einfach im *Wörterbuch* oder beim *Begriff des Tages* nach der Lösung für eine Frage nachgesehen wird. Zudem gewährt das Ausblenden der *Tab Bar* unter Umständen mehr Raum, um die Inhalte des Quiz darstellen zu können und zusätzliches Scrollen seitens des Nutzer zu umgehen.

Für die Orientierung des Nutzers steht trotz ausgeblendeter *Tab Bar* weiterhin die *Navigation Bar* zu Verfügung, die anzeigt, dass man sich gerade im Quiz befindet und durch Buttons die Möglichkeit gibt, das Quiz jederzeit abubrechen und damit die *Tab Bar* wieder einzublenden.

## 2.4 Didaktisches Konzept

...