

Abschlussprüfung Sommer 2018

Fachinformatiker für Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

## **Chat Bot**

### **Suchassistent für Soziale Netzwerke**

Abgabedatum: München, den 15.05.2018

#### **Prüfungsbewerber:**

Florian Huschka

#### **Ausbildungsbetrieb:**

JobNinja GmbH

Schwanthaler Straße 76

80336 München



## Inhaltsverzeichnis

---

### Inhaltsverzeichnis

Inhaltsverzeichnis.....	I
Abbildungsverzeichnis.....	III
Tabellenverzeichnis.....	IV
Verzeichnis der Listings.....	V
Abkürzungsverzeichnis.....	VI
Einleitung .....	7
Projektumfeld.....	7
Projektziel .....	7
Projektbegründung .....	7
Projektschnittstellen.....	7
Projektabgrenzung.....	7
Projektplanung .....	8
Projektphasen.....	8
Abweichungen vom Projektantrag.....	8
Ressourcenplanung .....	8
Entwicklungsprozess .....	8
Analysephase.....	9
Ist-Analyse .....	9
Wirtschaftlichkeitsanalyse .....	9
Make or Buy-Entscheidung .....	9
.....	9
Projektkosten .....	9
Amortisationsdauer .....	10
Nutzwertanalyse .....	10
Anwendungsfälle .....	10
Entwurfsphase .....	11
Zielplattform.....	11
Architekturdesign .....	11
Implementierungsphase .....	13
Implementierungen innerhalb des Frameworks.....	13
Starten des Dialogs mit dem Bot.....	14
Abnahmephase .....	19
Einführungsphase .....	19
Dokumentation .....	19
Fazit .....	19
Soll-/Ist-Vergleich.....	19

## **Inhaltsverzeichnis**

---

Lessons Learned .....	19
Ausblick .....	20
Literaturverzeichnis .....	20
Eidesstattliche Erklärung .....	21
Anhang.....	i
A1    Detaillierte Zeitplanung.....	i
A2    Aktivitätsdiagramm .....	ii
A3    Screenshots der Anwendung .....	iii

## Abbildungsverzeichnis

---

### Abbildungsverzeichnis

Abbildung 1: Aktivitätsdiagramm .....	Anhang A2
Abbildung 2: Screenshot 1.....	Anhang A3
Abbildung 3: Screenshot 2.....	Anhang A3
Abbildung 4: Screenshot 3.....	Anhang A3
Abbildung 5: Screenshot 4.....	Anhang A3
Abbildung 6: Screenshot 5.....	Anhang A3

## Tabellenverzeichnis

---

### Tabellenverzeichnis

Tabelle 1: Grobe Zeitplanung .....	8
Tabelle 2: Brutto Arbeitsleistung .....	9
Tabelle 3: Laufende Kosten / Monat .....	9
Tabelle 4: Nutzenrechnung .....	19
Tabelle 5: Detaillierte Zeitplanung .....	Anhang A1

## Verzeichnis der Listings

---

### Verzeichnis der Listings

Listing 1: Grundkonfiguration.....	<b>Fehler! Textmarke nicht definiert.</b>	13
Listing 2: Erste Interaktion des Users .....		14
Listing 3: Aufforderung zur Eingabe .....		14
Listing 4: Eingabeaufforderung der Stadt.....		15
Listing 5: Erstellung der Herocards und des Herocard-Carousell.....		16
Listing 6: Hinzufügen der funktion Buttons "Details" und "Jetzt bewerben".....		17
Listing 7: Präsentation des geparsten Ergebnisses im Messengerfenster.....		18

## Abkürzungsverzeichnis

API .....	<i>Application Programming Interface</i>
CEO .....	<i>Chief Executive Officer</i>
CTO .....	<i>Chief Technology Officer</i>
ERM .....	<i>Entity Relationship Model</i>
GUI .....	<i>Graphical User Interface</i>
HTML .....	<i>Hypertext Markup Language</i>
SQL .....	<i>Structured Query Language</i>

## Einleitung

### Projektumfeld

Die JobNinja GmbH ist im Bereich der Jobvermittlung tätig, und hat den Markt der Personalbeschaffung mit ihrem Produkt „Tinder für Jobs“ revolutioniert. Die Jobsuche kann nach den konkreten Jobwünschen angepasst werden. Über die One-Click- Bewerbung auf der JobNinja-Webseite bzw. über einen Rechts - Swipe in der JobNinja - App kann man sich direkt bewerben! Die bereits in der Datenbank gespeicherten Bewerberinformationen werden sofort an den Arbeitgeber übermittelt.

(JobNinja GmbH, 2017)

### Projektziel

Nutzer sind heute mehr als je zuvor in Apps wie Facebook Messenger, WhatsApp, Telegram oder Viber aktiv. Tatsächlich verbringen Nutzer weltweit unlängst mehr Zeit in den führenden Messaging-Apps als in den 4 führenden sozialen Netzwerken. Und dieser Vorsprung wächst immer stärker an. Um die Kunden bei der „mobile“ Jobsuche zu unterstützen wurde überlegt welche Möglichkeiten umsetzbar wären.

### Projektbegründung

Die JobNinja GmbH hat sich entschieden einen eigenen Chat Bot zu entwickeln. In diesem Projekt geht es um die Basis - Realisierung eines Chat Assistenz Systems das mit den minimalsten Eingaben des Kunden wie Name, Job Titel und Stadt eine personalisiertes Ergebnis ausgibt.

### Projektschnittstellen

Der Bot greift auf die interne API der JobNinja Job-Börse zu um nach Ergebnissen zu suchen.

Weitere Schnittstellen ist das Microsoft Bot Framework sowie die Cloud-Computing-Plattform Microsoft Azure und der Facebook Messenger auf den über einen Developer Account von Facebook zugegriffen werden kann.

### Projektabgrenzung

In diesem Projekt geht es nur um die Grundfunktionen des Chat-Bot, der in der Lage sein soll auf „Keywords“ zu reagieren und den Nutzer durch die Jobsuche im Messenger führt.



## Projektplanung

In der Projektplanung werden benötigte Ressourcen, Phasen und der Ablauf des Entwicklungsprozesses geplant

### Projektphasen

Das Projekt wurde während der regulären Tagesarbeitszeit im Zeitraum vom 30.04.2018 bis zum 11.05.2018 erstellt und dokumentiert.

In Tabelle 1 ist eine grobe Zeitplanung kurz zusammengefasst, eine detaillierte Zeitplanung ist in Tabelle 4 im Anhang 1 zu finden.

Projektphase	Geplante Zeit
Analyse	4 h
Entwurf	9 h
Implementierung	40 h
Abnahme & Testung	7 h
Dokumentation	10 h
<b>Gesamt</b>	<b>70 h</b>

Tabelle 1: Grobe Zeitplanung

### Abweichungen vom Projektantrag

Aufgrund der kurzen Projektzeit von 70 Stunden wurde auf die Erstellung eines Lasten und Pflichtenhefts verzichtet da der Zeitrahmen sonst nicht einzuhalten gewesen wäre.

### Ressourcenplanung

Für die Erstellung des Projekts wird ein Schreibtisch Arbeitsplatz mit einem Laptop (HP Elite-Book 8560p mit Windows 7 x64) sowie ein Internetzugang benötigt. Falls erforderlich besteht die Möglichkeit auf das fachliche Knowhow des CTO's Hr. Laurent Meyer (Full Stack Developer) zurück zu greifen um alle Anforderungen an das Projekt zu erfüllen.

### Entwicklungsprozess

Die Entwicklung erfolgte nach den Methoden der agilen Softwareentwicklung, diese Methoden sind bei JobNinja Standard und bedurften keiner weiteren Schulung oder Unterweisung.

## Analysephase

Nach Abschluss der Planungsphase ist es nun in der Analysephase wichtig den Ist – Zustand zu ermitteln und wirtschaftliche Aspekte in den Fokus zu rücken.

### Ist-Analyse

Bislang existierte nur eine Textversion in Form einer Benutzeranleitung für die Benutzung der Jobbörse die eigentlich „intuitiv“ sein soll. Diese Variante ist nun nicht mehr aktuell und soll durch eine Interaktive Version in Form des Bot ersetzt werden.

### Wirtschaftlichkeitsanalyse

#### Make or Buy-Entscheidung

Aufgrund sehr spezieller Anforderungen an den Bot stand eine Make or Buy Entscheidung nicht zur Debatte.

#### Projektkosten

Bezeichnung	Name	Vorgang	Zeit	Kosten / Stunde	Kosten
Arbeitslohn* Junior Entwickler	Florian Huschka	Entwicklung	70h	40,00 €	2800,00 €
Arbeitslohn* Projektleiter		Beratung & Hilfe	20h	100,00 €	2000,00 €

Tabelle 2: Brutto  
Arbeitsleistung

**Brutto Arbeits-  
leistung**

**4.800,00 €**

Bezeichnung	Kosten / Monat
Wartungskosten pauschal	150,00 €
Microsoft Azure Server	40,00 €
Microsoft Bot Framework	0,00 €
Microsoft Visual Studio Code	0,00 €
Summe laufende Kosten	190,00 €

Tabelle 3: Laufende Kosten / Monat

## Amortisationsdauer

Monetärer Vorteil

Einsparung von durchschnittlich 10 Stunden im Monat Arbeitszeit des CEO's der User angeleitet hat oder Fragen beantwortet hat.

Bei einer Zeiteinsparung von 0,5 Stunden am Tag für den CEO und 220 Arbeitstagen im Jahr ergibt sich eine gesamte Zeiteinsparung von:

$$0,5h \cdot 220 \frac{\text{Tage}}{\text{Jahr}} = 110 \frac{h}{\text{Jahr}}$$

Dadurch ergibt sich eine jährliche Einsparung von:

$$110h \cdot 135,00 \frac{\text{€}}{h} = 16.200,00 \text{ €}$$

Die Amortisationszeit beträgt also:

$$\text{Amortisationszeit} = \frac{\text{Anschaffungsausgabe}}{\text{durchschnittlicher Rückfluss}}$$

$$\frac{4.800,00\text{€}}{(1.350,00\text{€} - 190,00\text{€})} \approx 4,137 \text{ Monate} \approx 4 \text{ Monate 1 Woche}$$

Bezeichnung	Name	Brutto/Std.	Stunden/Mon.	Summe Monat	Summe Jahr
Arbeitsleistung CEO**	Mircea Popa	135,00 €	10	1.350,00 €	16.200,00 €

Tabelle 4: Nutzenrechnung

## Nutzwertanalyse

- \*\* direkter Nutzen aufgrund gesparter Arbeitszeit durch automatische Beantwortung von Kunden Rückfragen.
- \*\* indirekte Nutzen monetär nicht berücksichtigt: Social Media Marketing und Brand Awareness aufgrund der innovativen Lösung.

## Anwendungsfälle

- aktiv Job-Suchender sucht innerhalb des Messengers, ChatBot übernimmt die Anfrage und führt den Interessenten durch den Such-Prozess.
- User stellt eine Frage, diese werden anhand von „Keywords“ ausgewertet und entsprechend mit vorgefertigten Antworten beantwortet.

Ein Aktivitätsdiagramm für den ersten Fall, befindet sich im **Anhang A2**.

## Entwurfsphase

### Zielplattform

- Als Ziel für den Chatbot wurde der Messenger von Facebook gewählt da dieser über die nötige Unterstützung und API verfügt.

### Architekturdesign

- **MS Bot Framework / Azure Bot:** Service bietet Tools zum Erstellen, Testen, Bereitstellen und Verwalten intelligenter Bots an einem zentralen Ort. Durch das modulare und erweiterbare Framework, das vom SDK bereitgestellt wird, können Entwickler mithilfe von Vorlagen Bots erstellen, die Sprache, Sprachverständnis, Fragen und Antworten und mehr bieten. (Microsoft Corporation, 2018) übersetzt durch Florian Huschka.
- **MS Azure :** Azure ist eine umfangreiche Sammlung integrierter Cloud-Dienste, mit denen Entwickler und IT-Profis über unser globales Netzwerk aus Rechenzentren Anwendungen erstellen, bereitstellen und verwalten. Integrierte Tools, DevOps und ein Marketplace unterstützen Sie bei der effizienten Erstellung – von einfachen mobilen Apps bis hin zu Lösungen auf Internetebene. (Microsoft Corporation, 2018)
- **Node.js :** Node.js® ist eine JavaScript-Laufzeitumgebung, die auf Chromes V8 JavaScript-Engine basiert. Durch ein Event-basiertes, blockierungsfreies I/O-Modell ist Node.js schlank und effizient. Mit npm hat Node.js das größte Ökosystem für Open Source Bibliotheken der Welt. (Node.js Foundation, 2018)
- **Restify :** Framework für die implementierung von RESTful Web-Services in Node.js. Optimiert für die Erstellung semantisch korrekter RESTful-Web-Services, die für den produktiven Einsatz in großem Umfang bereit sind. restify optimiert für Introspektion und Leistung und wird in einigen der größten Node.js-Bereitstellungen auf der Welt verwendet. (restify.com, 2018)

- **Request-promise:** Der vereinfachte http-Anfrage-Client mit promise Unterstützung von Bluebird. Promises sind primitive neben Prozesse mit nachgewiesener Erfolgsbilanz und Sprachintegration in den meisten modernen Programmiersprachen. Sie werden seit den 80er Jahren intensiv verwendet. (Bluebird, 2016)
- **Bot Builder SDK :** Das Bot Builder SDK für Node.js ist ein Framework für die Entwicklung von Bots. Es ist einfach zu verwenden und modelliert Frameworks wie Express & Restify, um JavaScript-Entwicklern eine vertraute Möglichkeit zu bieten, Bots zu schreiben. (Microsoft Corporation, 2018)
- **Entwurf der Benutzeroberfläche:**  
Die Benutzeroberfläche musste nicht gesondert entworfen werden da sie vom Bot Framework zur Verfügung gestellt wird, und bei Facebook das bestehende Design des Messengers verwendet wird.  
Screenshot des Frameworks befindet sich in **Anhang A3**

## Implementierungsphase

### Implementierungen innerhalb des Frameworks

Grundkonfiguration mit Erstellung aller Variablen, Setup des Restify Servers sowie Verbindungsaufbau zu den Servern.

```
var rp = require("request-promise");
var builder = require("botbuilder");
var restify = require("restify");
const fs = require("fs");
var _ = require("lodash");

const PROD = false;
const BASE_URL = PROD
  ? "https://jnprod.herokuapp.com"
  : "https://jnstage.herokuapp.com";
var base64 = require("base-64");
var s3 = require("./s3");
// Setup Restify Server
var server = restify.createServer();
server.listen(process.env.port || process.env.PORT ||
3978, function() {
  console.log("%s listening to %s", server.name,
server.url);
});
var connector = new builder.ChatConnector({
  appId: process.env.MICROSOFT_APP_ID,
  appPassword: process.env.MICROSOFT_APP_PASSWORD
});
```

Listing 1: Grundkonfiguration

## Starten des Dialogs mit dem Bot

Der Bot ist aufgrund der Bestimmungen für Chat-Bots so konfiguriert das der erste Kontakt vom User auszugehen hat. Somit reagiert der Bot auf jegliche Ansprache mit einer Begrüßung.

```
// First User Handle await
server.post("/api/messages", connector.listen());

var bot = new builder.UniversalBot(connector, [
    function(session) {
        session.send("Hey wir von JobNinja helfen dir bei der Jobsuche!");
        session.beginDialog("askForjob");
    }
]);
```

Listing 2: warten auf Interaktion des Users

## Interaktion Bot – User

„Suche nach Job-Titel oder Bezeichnung“

```
// Dialog to askt 4 Job Keyword
bot.dialog("askForjob", [
    function(session) {
        session.dialogData.query = {};
        builder.Prompts.text(
            session,
            "Gib uns ein Stichwort nach welchem Job wir für dich suchen sollen"
        );
    },
    ],
```

Listing 3: Aufforderung zur Eingabe

Eingabeaufforderung der Stadt in dem nach dem Job gesucht werden soll. Falls nichts eingegeben wird, sucht der Bot automatisch Bundesweit nach Stellenangeboten die den oberen Suchbegriff im Titel oder im Beschreibungstext enthalten.

```
// Query of the place in it shall be looked for a job
function(session, results, next) {
  if (results.response) {
    session.dialogData.query.keyword = results.response;
    builder.Prompts.text(
      session,
      `Wo sollen wir für Dich nach ${session.dialogData.query
        .keyword} suchen?`
    );
  } else {
    next();
  }
},
```

Listing 4: Eingabe der Stadt



## Erzeugen eines „Herocard – Karusell“ für die Präsentation der Ergebnisse

```
//Output the found result in a Hero Card Carousel

function(session, results) {
  if (results.response) {
    session.dialogData.query.location_label = results.response;

    session.send(
      `Wir haben in ${session.dialogData.query.location_label} nach
      ${session
        .dialogData.query.keyword} gesucht!`
    );
    // Create HeroCard
    var options = {
      uri: `${BASE_URL}/v1/b2c/jobs?keywords=${encodeURIComponent(
        session.dialogData.query.keyword
      )}&location_label=${encodeURIComponent(
        session.dialogData.query.location_label
      )}&per_page=10`,
      json: true
    };
    rp(options).then(res => {
      var attachments = res.jobs.map(e => {
        return (
          new builder.HeroCard(session)
            .title(e.title)
            .subtitle(e.city)
            .text(e.description.substring(0, 0))
            .images([builder.CardImage.create(session, e.picture_url)])
        )
      })
    })
  }
}
```

Listing 5: Erstellen des Herocard-Carousel

## Einfügen der Buttons „Detail“ und „Jetzt in 30 Sekunden Bewerben“

```
// Button "Details"

        .buttons([

            builder.CardAction.openUrl(

                session,

                `https://jobninja.com/jobs/${e.id}?utm_source=chat-
                bot&utm_campaign=chatbot`,

                "Details"

            ),

// Button "In 30 Sekunden bewerben"

            builder.CardAction.imBack(

                session,

                `Ich will mich für den Job ${e.title} bewerben (id: ${e.id}).`,

                "Jetzt in 30 Sekunden Bewerben"

            )

        ])

    );

});
```

Listing 6: Hinzufügen von Funktion Buttons

## Präsentation der Suchergebnisse innerhalb des oben erstellten „Herocard-Carousel“

```
// Results of Search

var msg = new builder.Message(session);

if (res.total_jobs > 0) {

    msg.attachmentLayout(builder.AttachmentLayout.carousel);

    msg.attachments(attachments);

    session.send(msg).endDialog();

    session.send(

        `Wir haben ${res.total_jobs} Jobangebote für Dich gefunden,
        Details zum Job erfährst Du, wenn du auf die entsprechende Anzeige
        klickst!`

    );

} else {

    // No Results

    session.send(

        "Oh, Es tut uns Leid, aber wir haben heute keine Jobs für
        dich, möchtest Du eine neue Suche starten?"

    );

}
```

Listing 7: Ergebnisse geparkt in Herocards

Abgrenzung des Projekts, nach Präsentation der Ergebnisse endet diese Projektarbeit da sonst der zeitliche und inhaltliche Rahmen den Zeitplan übersteigen würde.

## Abnahmephase

Der Chatbot wurde durch die Entwickler während der gesamten Entwicklungszeit ständig dynamisch getestet, die dabei erzielten Ergebnisse wurden direkt in den Entwicklungsprozess übernommen.

Die Betaversion wurde durch das Jobninja Marketing und Sales Team auf Usability und User Experience getestet und positiv bewertet und abgenommen durch CEO Mircea Popa.

## Einführungsphase

Nach Abschluss der Abnahme und Testphase stand das Deploying auf den Azure Server auf der Agenda, dieses wurde automatisiert durchgeführt, die Verknüpfung mit dem Facebook Messenger wurde über die Facebook Messenger API im Developer Bereich von Facebook durchgeführt und erfolgte nach der Migration auf den Servern ebenfalls automatisiert.

## Dokumentation

Die Dokumentation wurde zum einen innerhalb des Codes mittels Kommentaren realisiert und diese Projektdokumentation erstellt.

## Fazit

### Soll-/Ist-Vergleich

Alle beteiligten Abteilungen sowie die Geschäftsführung sind mit dem Ergebnis der Entwicklung zufrieden

Phase	Geplant	Tatsächlich	Differenz
Analyse	4 h	5 h	+1 h
Entwurf	9 h	8 h	-1 h
Implementierung	40 h	35 h	-5 h
Abnahme	7 h	10 h	+3 h
Dokumentation	10 h	12 h	+2 h
<b>Gesamt</b>	<b>70 h</b>	<b>70 h</b>	

Tabelle 3: Soll-/Ist-Vergleich

### Lessons Learned

- Zeitmanagement
- Node.js
- Microsoft Bot Framework
- Microsoft Azure

## Ausblick

- In der Zukunft wird das Projekt weiter verfolgt und ausgebaut, es werden noch Komponenten wie Registrierung und „Jetzt Bewerben“ innerhalb des Messenger Fensters realisiert.

## Literaturverzeichnis

Bluebird 2016. *bluebirdjs.com*. [Online]  
Available at: [bluebirdjs.com](https://bluebirdjs.com)  
[Accessed 5 05 2018].

Microsoft Corporation, 2018. [Online]  
Available at: <https://docs.microsoft.com/en-us/azure/bot-service/bot-service-overview-introduction?view=azure-bot-service-3.0>  
[Accessed 09 05 2018].

Node.js Foundation, 2018. *nodejs.org*. [Online]  
Available at: [nodejs.org/de/](https://nodejs.org/de/)  
[Accessed 11 05 2018].

restify.com, 2018. *restify.com*. [Online]  
Available at: [restify.com](https://restify.com)  
[Accessed 30 04 2018].

## Eidesstattliche Erklärung

Ich, Florian Huschka | Prüflings Nummer 155 121 91 126, versichere hiermit, dass ich meine Dokumentation zur betrieblichen Projektarbeit mit dem Thema

*Chat Bot – Suchassistent für Soziale Netzwerke*

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

München, den 15.05.2018

---

FLORIAN HUSCHKA | PRÜFLINGS NUMMER

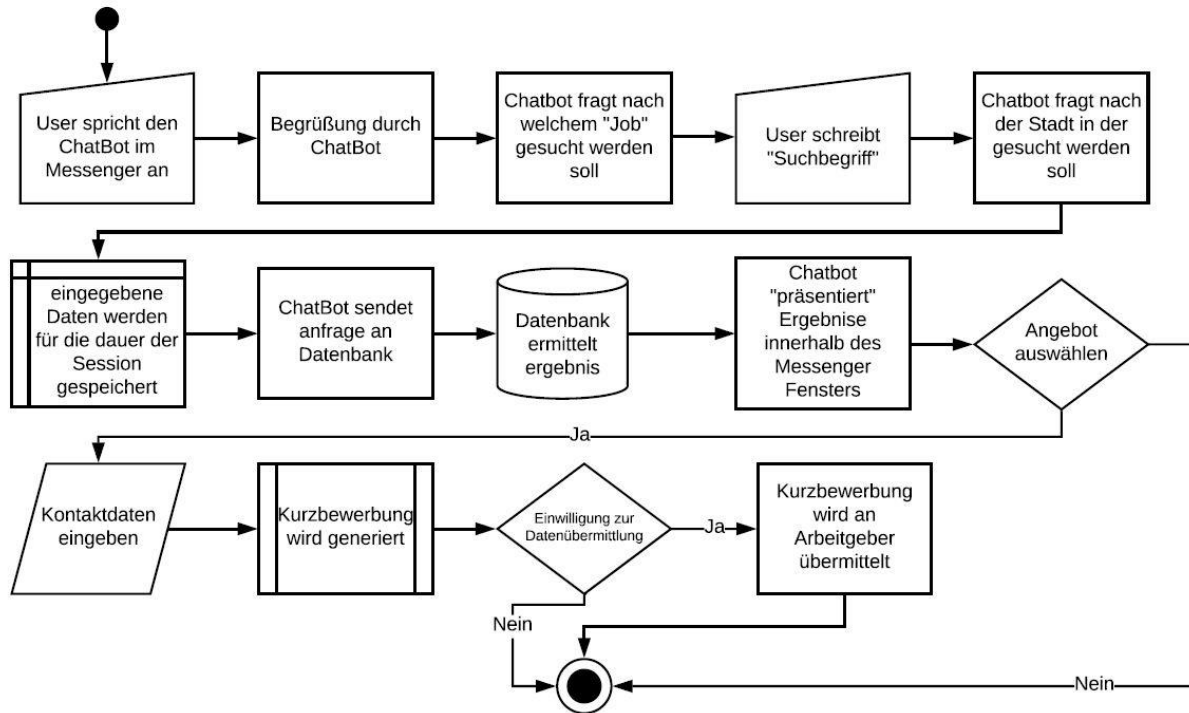
## Anhang

### A1 Detaillierte Zeitplanung

<b>Analysephase</b>	<b>5 h</b>
1. Analyse des Ist-Zustands	1 h
1.1. Fachgespräch mit der EDV-Abteilung	2 h
3. Erstellen eines Use-Case-Diagramms	2 h
<b>Entwurfsphase</b>	<b>8 h</b>
1. Prozessentwurf	3 h
2. ER-Modell erstellen	2 h
3. Erstellen von Datenverarbeitungskonzepten	1 h
5. Erstellen eines UML-Komponentendiagramms der Anwendung	2 h
<b>Implementierungsphase</b>	<b>35 h</b>
1. Installation Frameworks mit Komponenten	3 h
2. Umsetzung Wasserfalldialoge	5 h
3. Programmierung des Herocard Karusells	10 h
3.1. Import der Jobs aus der Datenbank (über API)	5 h
3.2. Parsen der Jobs in Herocards	3 h
3.3. implementierung des „Herocard Carousells“ in den Messenger	4 h
3.4. Implementierung der Buttons	2 h
4. Migration aller Komponenten	3 h
<b>Abnahmetest der Fachabteilung</b>	<b>10 h</b>
1. Abnahmetest der Fachabteilungen	10 h
<b>Erstellen der Dokumentation</b>	<b>12 h</b>
1. Erstellen der Benutzerdokumentation	1 h
2. Erstellen der Projektdokumentation	9 h
3. Programmdokumentation	2 h
<b>Gesamt</b>	<b>70 h</b>

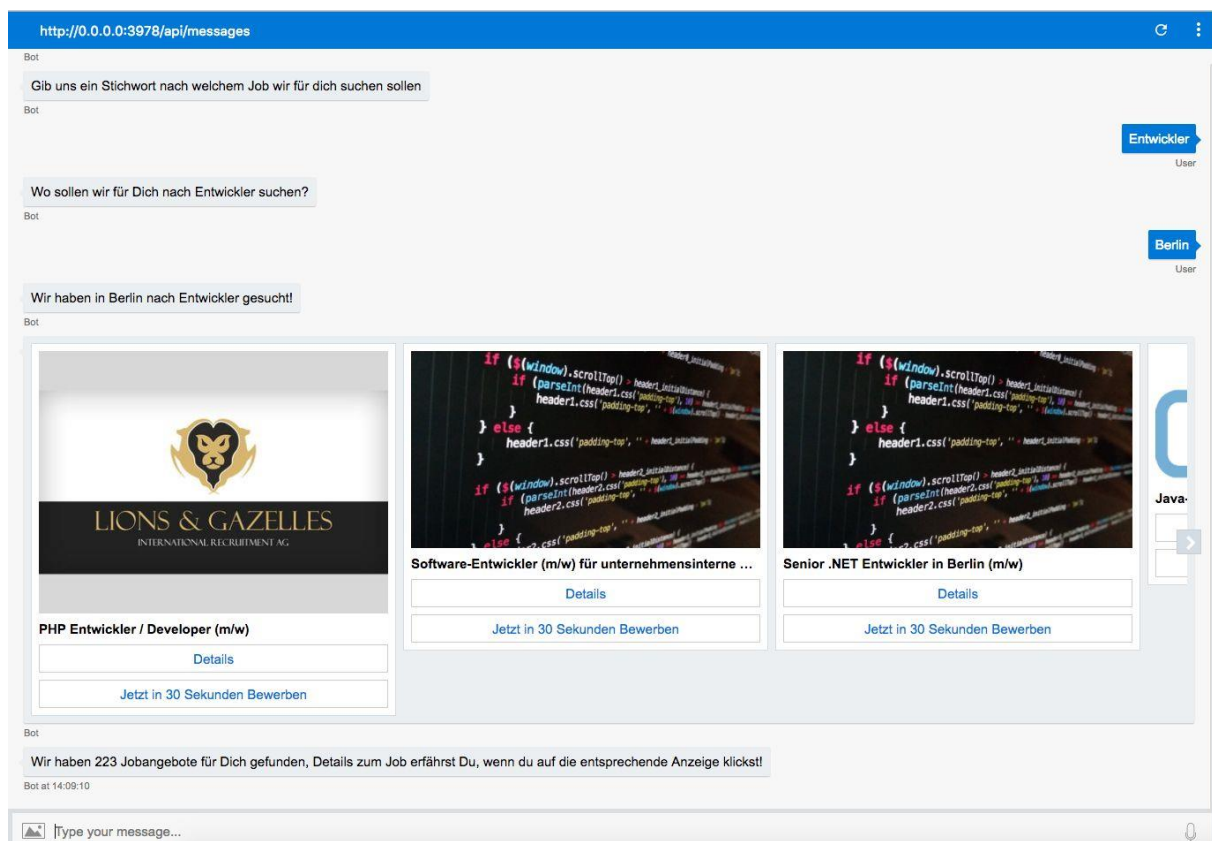
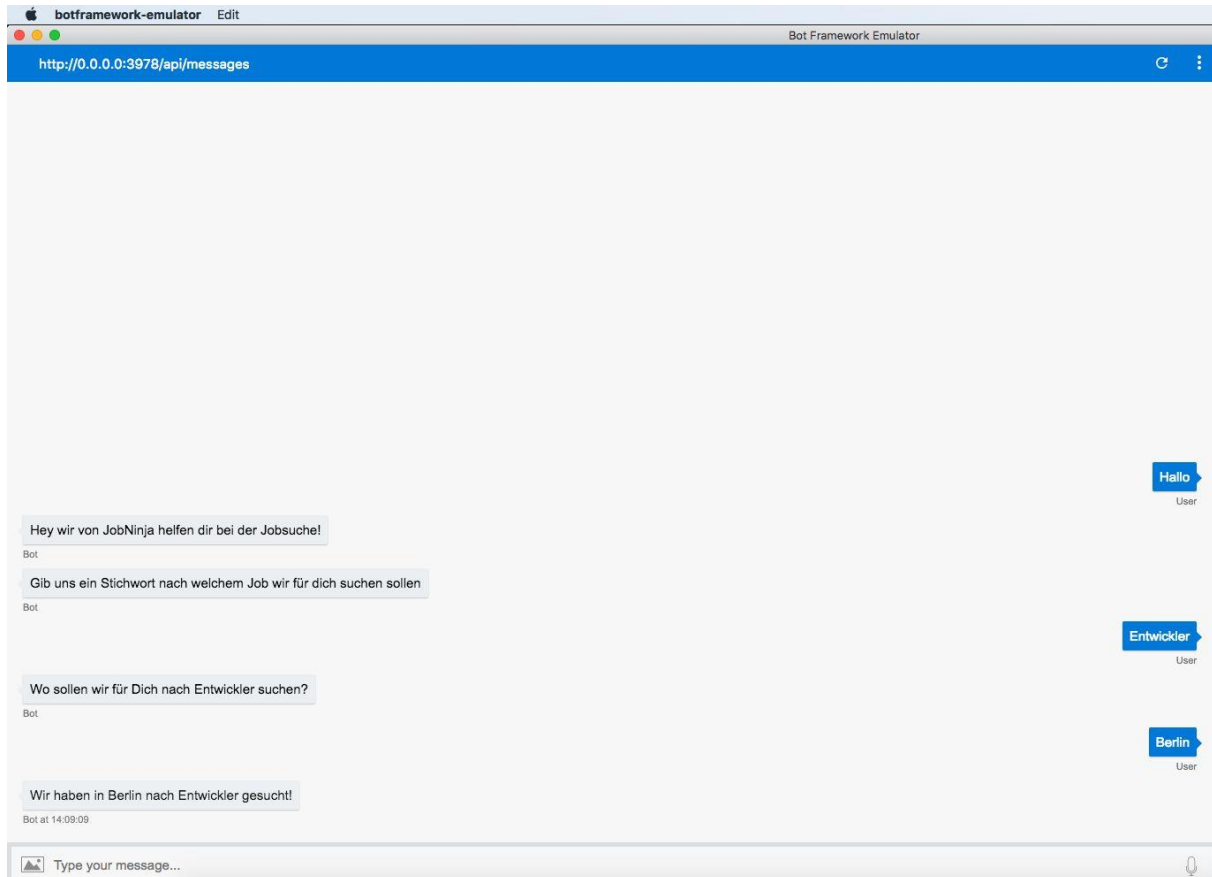
Tabelle 4: Detaillierte Zeitplanung

## A2 Aktivitätsdiagramm





## A3 Screenshots der Anwendung



http://0.0.0.0:3978/api/messages


Bot

Wo sollen wir für Dich nach Entwickler suchen?

Bot

Wir haben in Berlin nach Entwickler gesucht!

Bot




**LIONS & GAZELLES**  
INTERNATIONAL RECRUITMENT AG

**PHP Entwickler / Developer (m/w)**

Details


Jetzt in 30 Sekunden Bewerben



**Software-Entwickler (m/w) für unternehmensinterne ...**

Details

Jetzt in 30 Sekunden Bewerben



**Senior .NET Entwickler in Berlin (m/w)**

Details

Jetzt in 30 Sekunden Bewerben

Bot

Wir haben 223 Jobangebote für Dich gefunden, Details zum Job erfährst Du, wenn du auf die entsprechende Anzeige klickst!

Bot

Ich will mich für den Job PHP Entwickler / Developer (m/w) bewerben (id: 120292).

User

Damit wir Deine Bewerbung erfassen können, brauchen wir noch ein paar Angaben. Starten wir mit deinem Vornamen

Bot at 14:11:20

Type your message...

http://0.0.0.0:3978/api/messages

Bot

Wir haben 223 Jobangebote für Dich gefunden, Details zum Job erfährst Du, wenn du auf die entsprechende Anzeige klickst!

Bot

Ich will mich für den Job PHP Entwickler / Developer (m/w) bewerben (id: 120292).

User

Damit wir Deine Bewerbung erfassen können, brauchen wir noch ein paar Angaben. Starten wir mit deinem Vornamen

Bot

Servus Florian, jetzt brauchen wir noch deinen Nachnamen

Bot

Perfekt Florian, jetzt brauchen wir noch Deine E-Mail Adresse unter der wir Dich erreichen können

Bot

Danke Florian, jetzt brauchen wir noch eine Telefonnummer unter der Du erreichbar bist

Bot

Vielen Dank für deine Eingaben Florian  
Wir haben von dir folgende Daten erhalten :  
Vorname: Florian  
Nachname: Huschka  
E-Mail: flo@test.com  
Telefonnummer: 0123456789

Wenn das alles so in Ordnung ist und du konform mit unseren Datenschutzbestimmungen (<https://jobninja.com/docs/ds.pdf>) bist dann bestätige mit "ja" und du kannst Deinen Lebenslauf als Bild oder PDF hochladen

Bot at 14:11:56

Type your message...

http://0.0.0.0:3978/api/messages

Bot

Wir haben 223 Jobangebote für Dich gefunden, Details zum Job erfährst Du, wenn du auf die entsprechende Anzeige klickst!

Bot

Ich will mich für den Job PHP Entwickler / Developer (m/w) bewerben (id: 120292).

User

Damit wir Deine Bewerbung erfassen können, brauchen wir noch ein paar Angaben. Starten wir mit deinem Vornamen

Bot

Servus Florian, jetzt brauchen wir noch deinen Nachnamen

Bot

Perfekt Florian, jetzt brauchen wir noch Deine E-Mail Adresse unter der wir Dich erreichen können

Bot

Danke Florian, jetzt brauchen wir noch eine Telefonnummer unter der Du erreichbar bist


Bot

Vielen Dank für deine Eingaben Florian  
Wir haben von dir folgende Daten erhalten :  
Vorname: Florian  
Nachname: Huschka  
E-Mail: [flo@test.com](mailto:flo@test.com)  
Telefonnummer: 0123456789

User

Wenn das alles so in Ordnung ist und du konform mit unseren Datenschutzbestimmungen (<https://jobninja.com/docs/ds.pdf>) bist dann bestätige mit "ja" und du kannst Deinen Lebenslauf als Bild oder PDF hochladen

Bot at 14:11:56

 Type your message...

