DOMAIN MODEL und SEQUENZ DIAGRAMM

BTI7081 - Software Engineering and Design Frühlingssemester 2020

Berner Fachhochschule

Abteilung Technik und Informatik

Team Blau:
Timon Borter
Sven De Gasparo
Luca Mühlheim
Marc Muster
Elias Schmidhalter

Klienten: Prof. Urs Künzler Prof. Dr. Jürgen Vogel

DOMAIN MODEL und SEQUENZ DIAGRAM

BTI7081 - Software Engineering and Design Team Blau

Inhalt

Domain Model	3
Sequenz Diagramm	4
Klassen Model	5
Anhang	8
Abbildungsverzeichnis	8

Domain Model

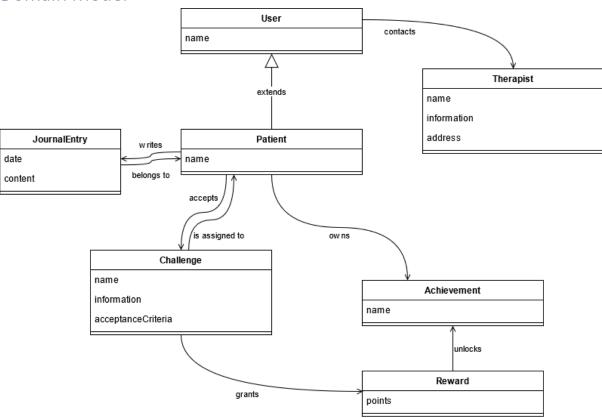


Abbildung 1: Domain Model

Sequenz Diagramm

Das Sequenzdiagramm zeigt den Prozess, in welchem der Patient eine Herausforderung annimmt, diese abschliesst und bewertet. Je nachdem, ob er eine gewisse Anzahl von Punkten erreicht hat erhält er eine Belohnung.

Die Ausgangslage dazu ist, dass der Patient bereits registriert und eingeloggt ist.

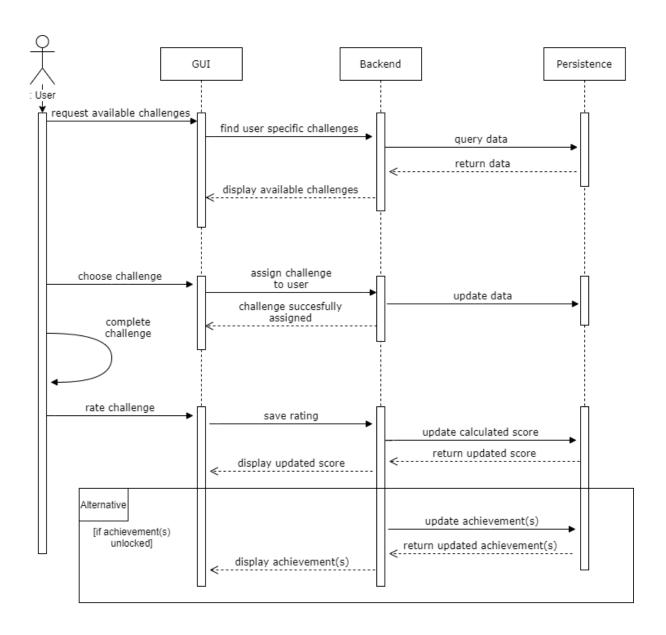


Abbildung 2: Sequenz Diagramm

Klassen Model

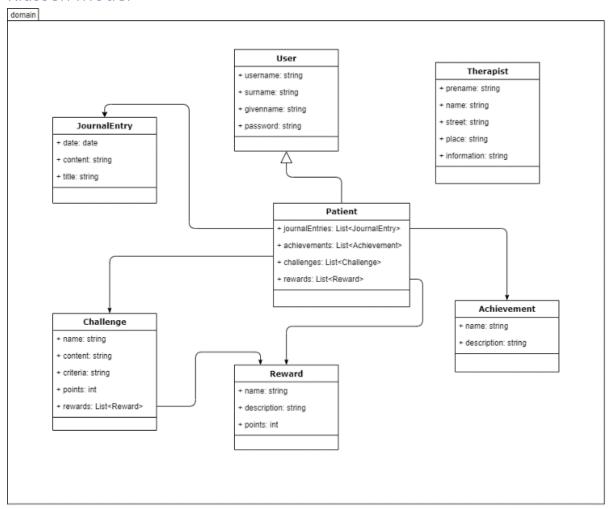


Abbildung 3: Klassen Diagramm, Domain

domain.repository PatientCRUDRepository TherapistCRUDRepository + save(patient: Patient): Patient + getAll(): List<Therapist> + login(username: string, password: string): void + save(therapist: Therapist): Therapist + logout(): void + getByld(id: int): void + register(username: string, password: string, surname: string, givenname: string): void ChallengeCRUDRepository + getAll(): List<Challenge> JournalCRUDRepository + save(challenge: Challenge): Challenge + getAll(): List<JournalEntry> + delete(id: int): void + save(journal: JournalEntry): Journal + delete(id: int): void AchievementCRUDRepository + getByPatient(patientId: int): List<JournalEntry> + getAll(): List<Achievement> + save(achievement: Achievement): Achievement RewardCRUDRepository + delete(id: int): void + getAll(): List<Reward> + save(reward: Reward): Reward + delete(id: int): void

Abbildung 4: Klassen Diagramm, Repository

BTI7081 - Software Engineering and Design Team Blau

domain.service PatientService TherapistService + userRepository: UserCRUDRepository + therapistRepository: TherapistCRUDRepository + getAll(): List<Therapist> + login(username: string, password: string): void + logout(): void + add(therapist: Therapist): void + register(username: string, password: string, surname: string, givenname: string): void + getByld(id: int): void + acceptChallenge(challenge: Challenge): void + closeChallenge(challenge: Challenge, successful: boolean): ChallengeService + challengeRepository: ChallengeCRUDRepository JournalService + getAll(): List<Challenge> + journalRepository: JournalCRUDRepository + add(challenge: Challenge): void + remove(id: int): void + getAll(): List<JournalEntry> + add(journal: JournalEntry): void + remove(id: int): void + getByPatient(patient: Patient): List<JournalEntry> AchievementService + achievementRepository: AchievementCRUDRepository RewardService + getAll(): List<Achievement> + rewardRepository: RewardCRUDRepository + add(achievement: Achievement): void + remove(id: int): void + getAll(): List<Reward> + add(reward: Reward): void + remove(id: int): void

Abbildung 5: Klassen Diagramm, Service

BTI7081 - Software Engineering and Design Team Blau

Anhang

Abbildungsverzeichnis

Abbildung 1: Domain Model	.3
Abbildung 2: Sequenz Diagramm	. 4
Abbildung 3: Klassen Diagramm, Domain	. 5
Abbildung 4: Klassen Diagramm, Repository	. 6
Abbildung 5: Klassen Diagramm, Service	. 7