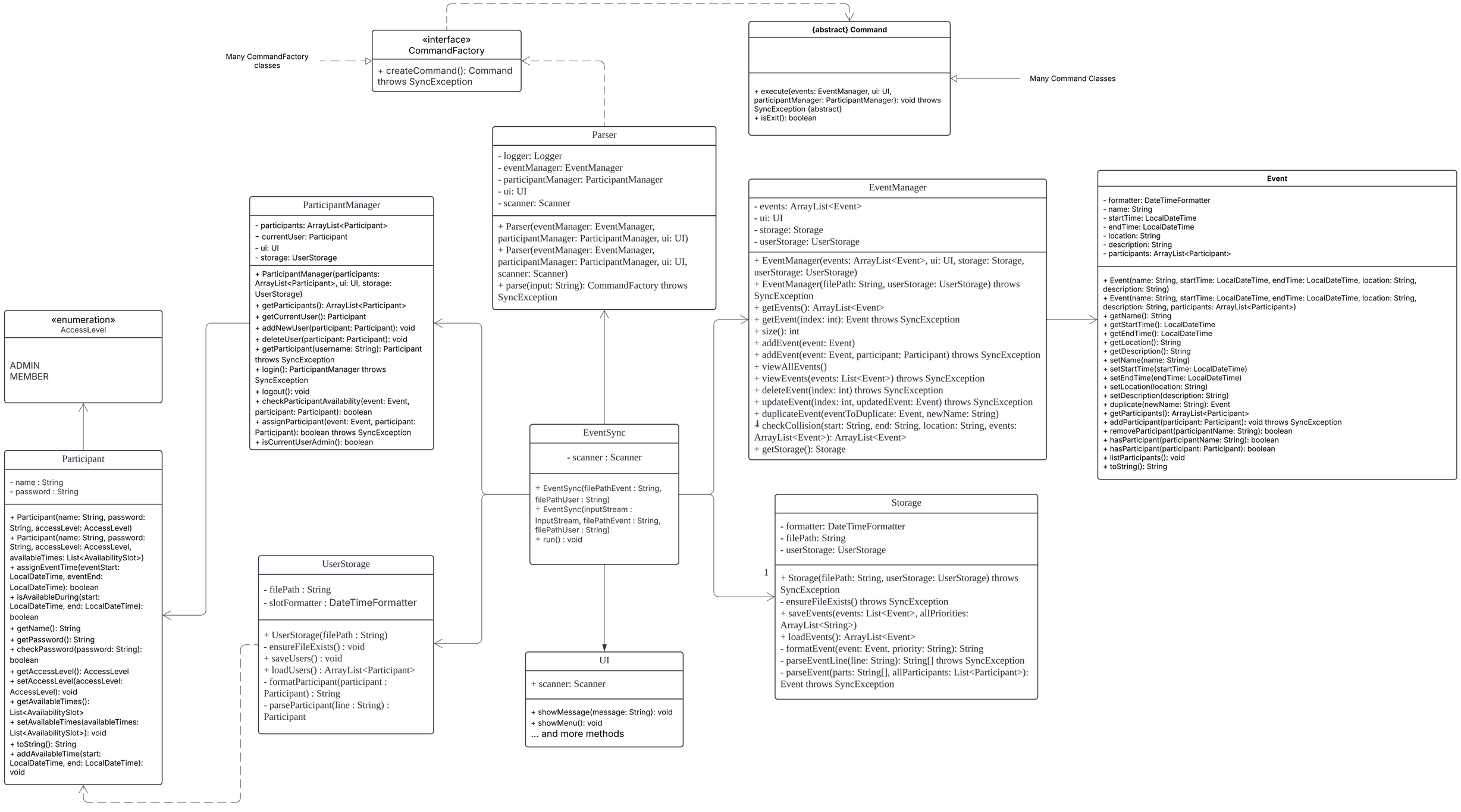
«interface»
CommandFactory
+ createCommand(): Command throws SyncException
Many CommandFactory classes
{abstract} Command
+ execute(events: EventManager, ui: UI, participantManager: ParticipantManager): void throws SyncException {abstract}
+ isExit(): boolean
Many Command Classes
Parser
- logger: Logger
- eventManager: EventManager
- participantManager: ParticipantManager
- ui: UI
- scanner: Scanner
+ Parser(eventManager: EventManager, participantManager: ParticipantManager, ui: UI)
+ Parser(eventManager: EventManager, participantManager: ParticipantManager, ui: UI, scanner: Scanner)
+ parse(input: String): CommandFactory throws SyncException
ParticipantManager
- participants: ArrayList<Participant>
- currentUser: Participant
- ui: UI
- storage: UserStorage
+ ParticipantManager(participants: ArrayList<Participant>, ui: UI, storage: UserStorage)
+ getParticipants(): ArrayList<Participant>
+ getCurrentUser(): Participant
+ addNewUser(participant: Participant): void
+ deleteUser(participant: Participant): void
+ getParticipant(username: String): Participant throws SyncException
+ login(): ParticipantManager throws SyncException
+ logout(): void
+ checkParticipantAvailability(event: Event, participant: Participant): boolean
+ assignParticipant(event: Event, participant: Participant): boolean throws SyncException
+ isCurrentUserAdmin(): boolean
EventManager
- events: ArrayList<Event>
- ui: UI
- storage: Storage
- userStorage: UserStorage
+ EventManager(events: ArrayList<Event>, ui: UI, storage: Storage, userStorage: UserStorage)
+ EventManager(filePath: String, userStorage: UserStorage) throws SyncException
+ getEvents(): ArrayList<Event>
+ getEvent(index: int): Event throws SyncException
+ size(): int
+ addEvent(event: Event)
+ addEvent(event: Event, participant: Participant) throws SyncException
+ viewAllEvents()
+ viewEvents(events: List<Event>) throws SyncException
+ deleteEvent(index: int) throws SyncException
+ updateEvent(index: int, updatedEvent: Event) throws SyncException
+ duplicateEvent(eventToDuplicate: Event, newName: String)
+ checkCollision(start: String, end: String, location: String, events: ArrayList<Event>): ArrayList<Event>
+ getStorage(): Storage
Event
- formatter: DateTimeFormatter
- name: String
- startTime: LocalDateTime
- endTime: LocalDateTime
- location: String
- description: String
- participants: ArrayList<Participant>
+ Event(name: String, startTime: LocalDateTime, endTime: LocalDateTime, location: String, description: String)
+ Event(name: String, startTime: LocalDateTime, endTime: LocalDateTime, location: String, description: String, participants: ArrayList<Participant>)
+ getName(): String
+ getStartTime(): LocalDateTime
+ getEndTime(): LocalDateTime
+ getLocation(): String
+ getDescription(): String
+ setName(name: String)
+ setStartTime(startTime: LocalDateTime)
+ setEndTime(endTime: LocalDateTime)
+ setLocation(location: String)
+ setDescription(description: String)
+ duplicate(newName: String): Event
+ getParticipants(): ArrayList<Participant>
+ addParticipant(participant: Participant): void throws SyncException
+ removeParticipant(participantName: String): boolean
+ hasParticipant(participantName: String): boolean
+ hasParticipant(participant: Participant): boolean
+ listParticipants(): void
+ toString(): String
«enumeration»
AccessLevel
ADMIN
MEMBER
Participant
- name : String
- password : String
+ Participant(name: String, password: String, accessLevel: AccessLevel)
+ Participant(name: String, password: String, accessLevel: AccessLevel, availableTimes: List<AvailabilitySlot>)
+ assignEventTime(eventStart: LocalDateTime, eventEnd: LocalDateTime): boolean
+ isAvailableDuring(start: LocalDateTime, end: LocalDateTime): boolean
+ getName(): String
+ getPassword(): String
+ checkPassword(password: String): boolean
+ getAccessLevel(): AccessLevel
+ setAccessLevel(accessLevel: AccessLevel): void
+ getAvailableTimes(): List<AvailabilitySlot>
+ setAvailableTimes(availableTimes: List<AvailabilitySlot>): void
+ toString(): String
+ addAvailableTime(start: LocalDateTime, end: LocalDateTime): void
EventSync
- scanner : Scanner
+ EventSync(filePathEvent : String, filePathUser : String)
+ EventSync(inputStream : InputStream, filePathEvent : String, filePathUser : String)
+ run() : void
UserStorage
- filePath : String
- slotFormatter : DateTimeFormatter
+ UserStorage(filePath : String)
- ensureFileExists() : void
+ saveUsers() : void
+ loadUsers() : ArrayList<Participant>
- formatParticipant(participant : Participant) : String
- parseParticipant(line : String) : Participant
UI
+ scanner: Scanner
+ showMessage(message: String): void
+ showMenu(): void
... and more methods
Storage
- formatter: DateTimeFormatter
- filePath: String
- userStorage: UserStorage
+ Storage(filePath: String, userStorage: UserStorage) throws SyncException
- ensureFileExists() throws SyncException
+ saveEvents(events: List<Event>, allPriorities: ArrayList<String>)
+ loadEvents(): ArrayList<Event>
- formatEvent(event: Event, priority: String): String
- parseEventLine(line: String): String[] throws SyncException
- parseEvent(parts: String[], allParticipants: List<Participant>): Event throws SyncException
1
1