

Application Development for Mobile and Ubiquitous Computing

Time Chicken

Final Presentation

GroupNo. 1

Team: Sebastian Fröstl, Christian Schäfer

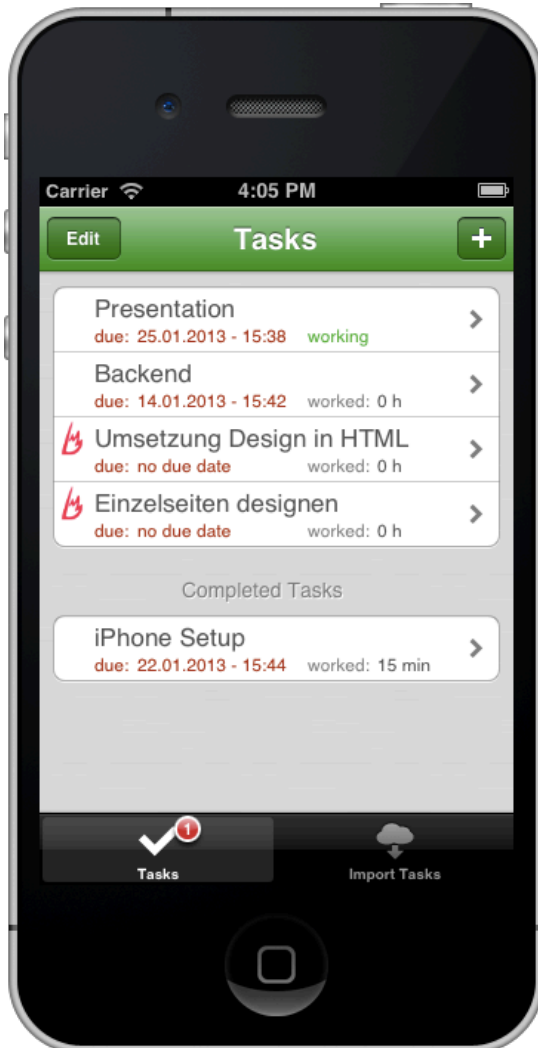


Time Chicken

is an App for managing Tasks and Time

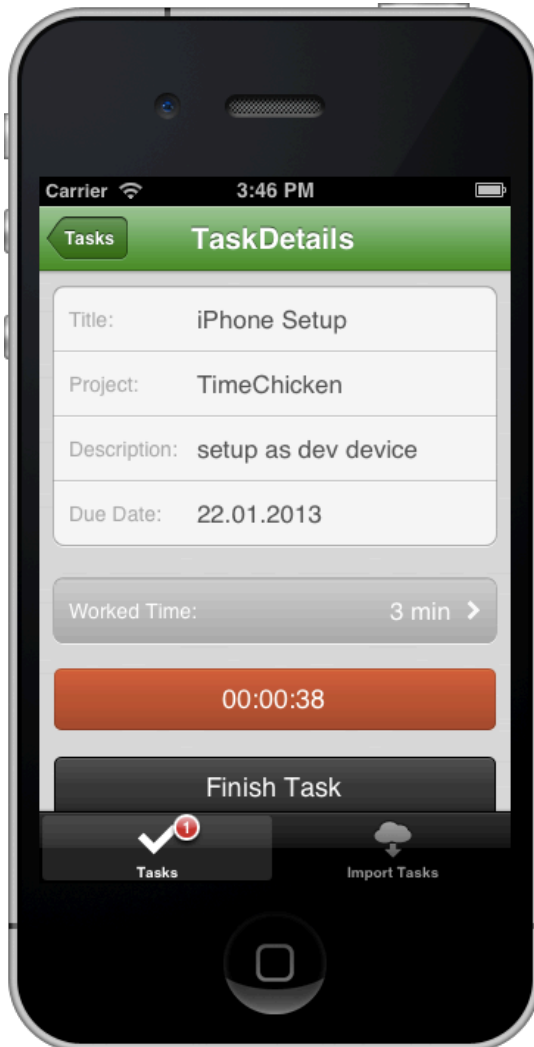
- Scenario 1: „I want an overview of my tasks.“
- Scenario 2: „I want to know how much working time I spend on a task.“
- Scenario 3: „I can sync TimeChicken with my Collaboration Plattform (Jira, OneSpark, ...).“





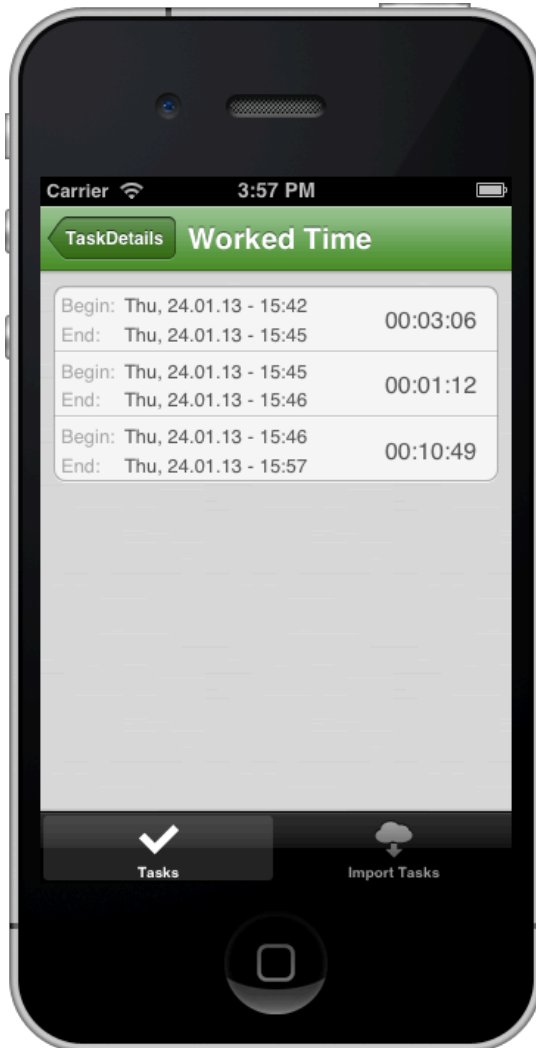
Task List

- Central Store for tasks as Singleton
- Task with all properties
- Persistent storage in NSDocument
- Two separate lists (Open/ Completed)
- List element with rough overview of a task
- Edit list (move and delete)
- Add new task



Task Details

- Edit attributes
- Start and stop timer and show on button
- Shows summary of worked time (All tracked time sessions)
- Set task as finished
- When timer is running show badge symbol on tab



Worked Time

- Overview of all tracked time sessions
- TimeSession - Object stores start and end time stamp and provides methods for displaying and calculating
- Stored in task



Import Tasks

- Different collaboration tools
- Interface for clients (protocol)
- Each client has to implement the methods to communicate with API

Import Detail View Choose Tasks for Import



- Lack of space
 - Very limited ability to display information
 - Timer button in task list cell was too much -> moved to detail view
- Customer Satisfaction
 - Easy navigation with tabs
 - Clear and limited functionality
 - Layout supports clarity (Common elements like lists with title)
 - Basic error handling in import task functionality
- Limited resources and energy
 - Network connection only on demand

- Taskmanagement with time tracking functions
- Import of tasks from two collaboration platforms
 - Jira, One Spark
- Software architecture is easy extendible to support other collaboration APIs.
 - New clients are implemented against common interface
 - New client handles specific backend access
- Customized views and buttons
- Deployment to devices for real usage testing

- Problems with „storyboard“ in collaborative work
- Git integration in xcode only limited features
- JSON mapping to attributes must be nil-save

- Sync imported tasks back to collaboration tool
- Notification for tasks with near due date
- Custom designed views
- Archive for deleted Tasks/ completed tasks
- Sorting and searching of task list
- More error handling
- Gestures

