

J2ME and BlackBerry Client Application for mobile advertisement network

IMPORTANT

Due to a governing confidentiality agreement, we've refrained from disclosing actual client and solution name. These have been changed appropriately, to more generic sounding terms and nomenclatures.

1. Introduction

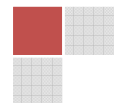
The J2ME/BlackBerry client application service for mobile ad network offers its users cash rewards for watching the ads on the mobile device. In order to enable the service, it is required to complete customer registration process and download the client application on the user's device.

This document examines the status of the client application from various perspectives and provides foundation for successful and timely completion of its development.

Industry:

Mobile Advertisement

Mobile Platform:



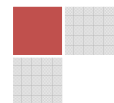
J2ME
BlackBerry

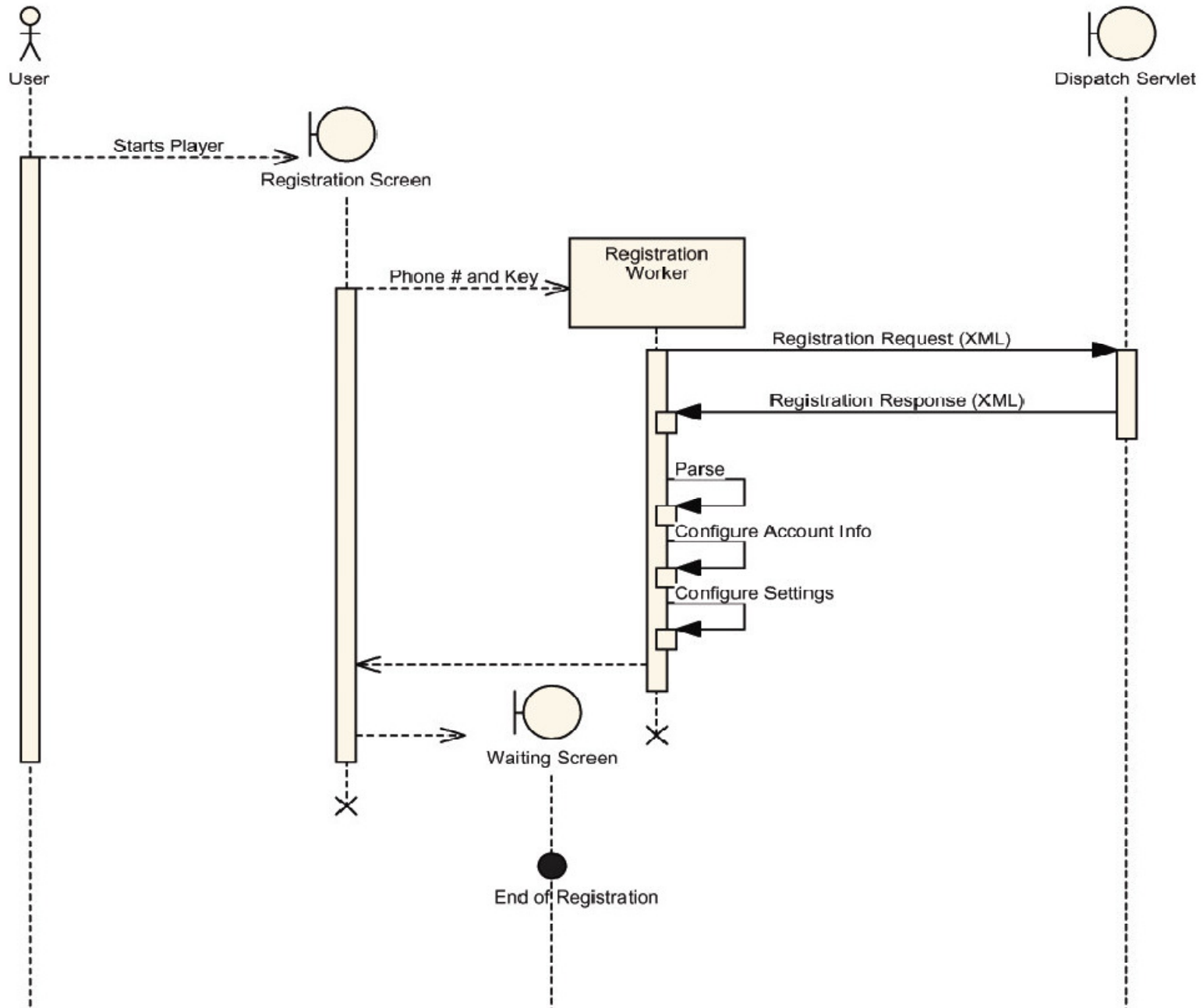
The application was developed and ported by us on all the latest J2ME and BlackBerry handsets.

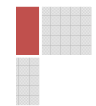
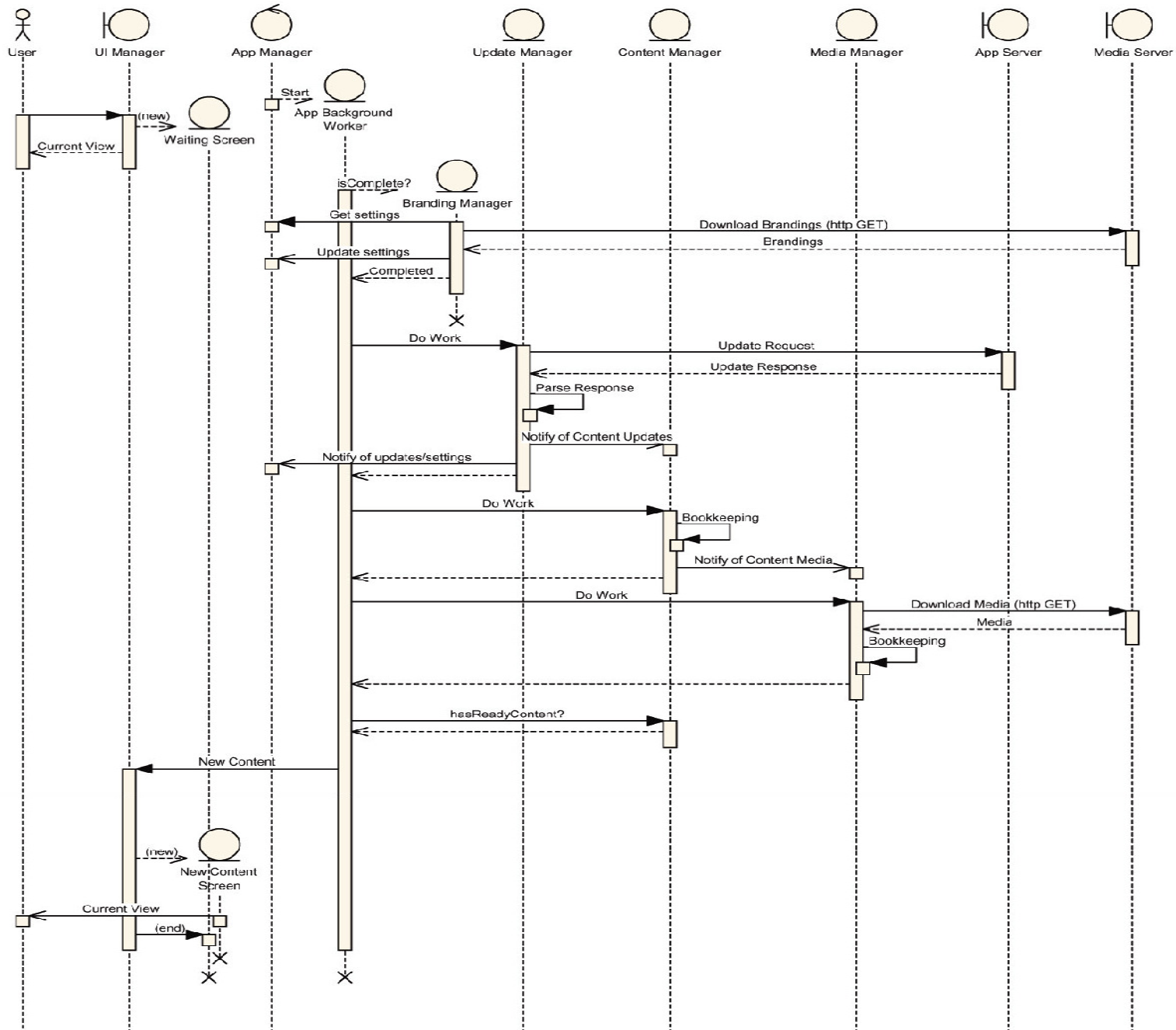
Run-Time Application Flow (Dynamic Analysis)

Two sequence diagrams below describe registration and general application flows.

www.RapidsoftTechnologies.com







Client–Server Protocol and Interactions

The following table summarizes all client-server interactions and protocols.

Application Flow Phase	Phase Summary	Protocol
Registration	<p>User account details (phone number and the registration key) and device properties are uploaded to the server as an XML doc. Server response is returned in XML format as well.</p> <p>Once the registration process is complete, 3 “Branding Media” video clips will be downloaded and saved in RMS.</p>	HTTP GET/POST
Check if a new content is available/upload user’s credits	<p>User account details, device properties and the current version ID are uploaded to the server as an XML doc. Server’s response (XML doc as well) indicates whether new ads are ready. It also includes the most recent client’s version number.</p>	HTTP GET/POST
Ads download	<p>Core application function. All clips are downloaded as a stream of bytes.</p>	HTTP GET

